TECHNICAL REPORT ARBRL-TR-02264

STRONG BLAST WAVE COMPUTER PROGRAMS

Aivars Celmiņš

September 1980

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

Destroy this report when it is no longer needed.
Do not return it to the originator.

Secondary distribution of this report by originating or sponsoring activity is prohibited.

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, Springfield, Virginia 22151.

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
TECHNICAL REPORT ARBRL-TR-02264

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Strong Blast Wave Computer Programs

5. TYPE OF REPORT & PERIOD COVERED
Technical Report

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Aivars Celmiņš

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
U.S. Army Ballistic Research Laboratory
ATTN: DRDAR-BLP
Aberdeen Proving Ground, MD 21005

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
RDTE 1L161102AH43

11. CONTROLLING OFFICE NAME AND ADDRESS
U.S. Army Armament Research & Development Command
U.S. Army Ballistic Research Laboratory
ATTN: DRDAR-BL
Aberdeen Proving Ground, MD 21005

12. REPORT DATE
September 1980

13. NUMBER OF PAGES
71

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Strong blast
Self similar solution
Computer programs
Spherical, cylindrical, planar symmetry
Mach-lines
Particle paths
Profiles
Histories

20. ABSTRACT (Continue on reverse side if necessary and identify by block number) jmk

This report describes a computer program package for the computation of the flow field within a strong blast bubble. The programs are based on Sedov-Laporte-Chang formulas and compute any of the following: shock front location and corresponding flow values, flow profiles at fixed times, flow histories at fixed distances, particle trajectories, and Mach-lines. The input and the output are in terms of dimensional quantities expressed in SI base units. The flow field can be calculated for a ratio of specific heats between one and seven, and for spherical, cylindrical, or planar symmetry.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF ILLUSTRATIONS

Figure Page

1. INTRODUCTION

Strong blast waves are produced in gaseous media by a sudden deposition of large amounts of energy in a relatively small region. Such events may be approximated mathematically by a point source wave. If one assumes in addition an ideal gas and an initial pressure negligible compared to the shock pressure, then the flow field can be described in closed form as a self-similar solution of the flow equations[1,2,3]. A derivation of such closed form solution, including spherical, cylindrical and planar blast waves, is given, e.g., by Sedov[4] and Rouse[5]. The solutions have been supplemented by Laporte and Chang[6,7] who derived closed form expressions also for the particle paths and for the Mach lines.

[1] *H.A. Bethe, K. Fuchs, J. von Neumann, R. Peierls, and W.G. Penny, U.S. Atomic Energy Commission Report AECD-2860 (1944).*

[2] *J.L. Taylor, "An Exact Solution of the Spherical Blast Wave Problem", Phil. Mag., Vol 46 (1955).*

[3] *R. Latter, "Similarity Solution for a Spherical Shock Wave", Journal of Applied Physics, Vol 26, No. 8 (1955).*

[4] *L.I. Sedov, "Similarity and Dimensional Methods in Mechanics", Academic Press, New York (1959).*

[5] *C.A. Rouse, "Theoretical Analysis of the Hydrodynamic Flow in Exploding Wire Phenomena, in "Exploding Wires", W.G. Chace and K.M. Howard, edts., Plenum Press, New York, pp. 227-263 (1959).*

[6] *O. Laporte and T.S. Chang, "Exact Expressions for Curved Characteristics Behind Strong Blast Waves", U.S. Army Ballistic Research Laboratory Contractor Report, BRL-CR-30 (January 1971).* (AD#722777)

[7] *O. Laporte and T.S. Chang, "Curved Characteristics Behind Blast Waves", Physics of Fluids, Vol 15, pp. 502-504 (1972).*

The closed form solutions are valuable approximations of physical phenomena, and they have been used as test cases for numerical solvers of flow governing equations[8,9]. The evaluation of the rather cumbersome formulas is, however, not trivial because the solution is expressed in parameter form. Therefore, also several tables of strong blast wave solutions have been published[4,8,10,11]. The use of such tables is, of course, limited to the particular set of parameters chosen by the authors of the tables. Also, because tables are expressed in form of dimensionless variables, they must be transformed before application to suit each particular case. Not all authors of the published tables provide sufficient and clear instructions as to how the transformations should be accomplished, nor are the tables of various authors standardized. Finally, the particle path and Mach line solutions of Laporte and Chang have not been tabulated at all.

In order to make the closed form solutions of strong blast waves easily available, we have developed a package of computer programs that produces the solution for any dimension and combination of parameters, within limits. The present report is a description of the programs and of their use.

In Section 2 of this report we shall summarize the formulas needed for flow computation. We shall not give a derivation of the formulas which can be obtained, e.g., from References 4 and 6, and which is based on the following assumptions:

1. the flowing medium is an ideal gas;

2. the initial pressure in the ambient gas can be neglected;

[8] *P.C. Chou and R.R. Karpp, "Solution of Blast Waves by the Method of Characteristics", Drexel Institute of Technology Report, DIT Report No. 125-7, (1965).*

[9] *J.A. Schmitt, "A Finite Element Method and Corresponding Pilot Computer Code for Hyperbolic Systems of Equations in Two Spatial Dimensions and Time Applied to Unsteady Gas Flows", U.S. Army Ballistic Research Laboratory Report, BRL-R-2017 (September 1977).* (AD#A045703)

[10] *N. Gerber and J.M. Bartos, "Tables of Cylindrical Blast Functions for $\gamma = 5/3$ and $\gamma = 7/5$", U.S. Army Ballistic Research Laboratory Report, BRL-R-1376 (October 1961).* (AD#663821)

[11] *J.W. Goresh and R.G. Dunn, "Tables of Blast Wave Parameters, I Spherical Explosions", Aerospace Research Laboratory Report, ARL-69-0011 (January 1969).*

3. the energy is released instantaneously at a point, in a line or in a plane.

For applications of the results to real life situations, the assumptions mean that the formulas may describe accurately events that are neither too far from the explosion (assumption 2), nor too close to it (assumption 3).

In Section 3 we shall describe the use of the computer programs for strong blast. Some examples of calculations will be presented in Section 4.

In Section 5 we derive quantitative conditions under which the strong blast solutions can provide approximate information about pressures generated by real life explosions in air.

2. THEORETICAL BACKGROUND

2.1. Shock Formulas

Let x_s be the location of the shock, i.e., the distance of the shock from the explosion, and let t be the time elapsed after the explosion. Using dimensional arguments[1-5] or group-theoretical discussions[6] one can derive the following relation between x_s and t:

$$x_s = K(n,\gamma) \left(\frac{E_o}{\rho_o}\right)^{1/(2+n)} t^{2/(2+n)} \quad . \tag{2.1}$$

In Eq. (2.1), $K(n,\gamma)$ is a proportionality factor that depends on the ratio of specific heats, γ, and on the dimension n of the event (n = 1, 2, or 3 for planar, cylindrical, or spherical symmetry, respectively), ρ_o is the density of the ambient gas, and E_o is the energy released, per unit area for a planar wave, per unit length for a cylindrical wave, and the total energy released for a spherical wave.

The proportionality factor $K(n,\gamma)$ is determined from the requirement that the total energy contained in the blast wave must be equal to the total energy released, i.e.,

$$E_o = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2})} \int_o^{x_s(t)} \rho(x,t) \left(u^2(x,t) + \frac{2}{\gamma-1} \; \frac{p(x,t)}{\rho(x,t)}\right) x^{n-1} \, dx. \tag{2.2}$$

In Eq. (2.2), $\rho(x,t)$ is the density of the gas, $u(x,t)$ is its particle velocity, and $p(x,t)$ is pressure. $\Gamma(\frac{n}{2})$ is the gamma function. It assumes the values $\sqrt{\pi}$, 1, and 1/2 $\sqrt{\pi}$ for n = 1,2, and 3, respectively.

If one substitutes in Eq. (2.2) for ρ, u, and p the self-similar solution formulas of Section 2.2. then the integral is found to be independent of time and proportional to E_o. It does, however, depend on n, γ and $K(n,\gamma)$. Therefore, Eq. (2.2) provides for each set of n and γ a corresponding value for the proportionality factor $K(n,\gamma)$. The integral has to be evaluated numerically. We shall return to the numerical quadrature in Section 3.2.

The pressure and the particle velocity at the shock are expressed most conveniently in terms of the shock velocity $U = dx_s(t)/dt$. The latter is, according to Eq. (2.1)

$$U(t) = \frac{2}{2+n} \quad \frac{x_s(t)}{t} \quad . \tag{2.3}$$

The particle velocity immediately behind the shock is

$$u_s(t) = \frac{2}{\gamma+1} U(t) \quad . \tag{2.4}$$

and the corresponding pressure is

$$p_s(t) = \frac{2}{\gamma+1} \quad \rho_o \; U^2(t) \quad , \tag{2.5}$$

where ρ_o is the density of the ambient gas. The density immediately behind the shock is

$$\rho_s = \frac{\gamma+1}{\gamma-1} \rho_o \quad . \tag{2.6}$$

We reiterate that Eqs. (2.3) through (2.6) are "strong shock" relations, i.e., they are based on the assumption that the ambient pressure p_o is negligible compared to the shock pressure p_s.

2.2. Flow Field Formulas

The closed form expressions for the flow field behind a strong blast are given, e.g., in References 4, 5, 6, and 8. In this report, we formulate the solution in a somewhat simpler but algebraically equivalent form.

The formulas express the distance, particle velocity, pressure, and density in terms of time and a dimensionless parameter v. The expressions are valid for ratios of specific heat that satisfy the conditions

$$1 < \gamma < 7 \quad \text{and} \quad \gamma \neq 2 \; . \tag{2.7}$$

The case $\gamma = 2$ requires a special treatment because some of the formulas become singular for that value of γ. The solution in this special case is simpler than the general solution[12], but we have not included the special solution in our computer programs.

The formulas for the flow variables are

$$x = x_s(t) \cdot y(v) \; , \tag{2.8}$$

$$u = u_s(t) \cdot \frac{v}{v_2} \cdot y(v) \; , \tag{2.9}$$

$$p = p_s(t) \cdot g(v) \; , \tag{2.10}$$

$$\rho = \rho_s \cdot h(v) \; . \tag{2.11}$$

The time functions and ρ_s, appearing in Eqs. (2.8) through (2.11) are defined in Section 2.1. The functions y(v), g(v), and h(v) are defined as follows:

$$y(v) = \left(\frac{v}{v_2}\right)^{-C_1} \left(\frac{v-v_1}{v_2-v_1}\right)^{C_2} \left(\frac{1-av}{1-av_2}\right)^{C_3} , \tag{2.12}$$

$$g(v) = \left(\frac{v}{v_2}\right)^{nC_1} \left(\frac{v_1\gamma-v}{v_1\gamma-v_2}\right)^{C_4} \left(\frac{1-av}{1-av_2}\right)^{C_5} , \tag{2.13}$$

$$h(v) = \left(\frac{v-v_1}{v_2-v_1}\right)^{1-2C_2} \left(\frac{v_1\gamma-v}{v_1\gamma-v_2}\right)^{C_4-1} \left(\frac{1-av}{1-av_2}\right)^{C_5-2C_3} . \tag{2.14}$$

[12] *A. Sakurai, "Blast Wave Theory", Mathematics Research Center Technical Summary Report 497 (September 1964).*

The dimensionless parameter v varies between

$$\left.\begin{aligned} v_1 &= \frac{2}{(n+2)\gamma} \\ &\text{and} \\ v_2 &= \frac{4}{(n+2)(\gamma+1)} \end{aligned}\right\} \qquad (2.15)$$

The value $v = v_1$ corresponds to the distance $x = 0$ from the explosion, and the value $v = v_2$ corresponds to a point on the shock.

The other constants in Eqs. (2.12) through (2.14) are

$$a = 1 + \frac{n}{2}(\gamma-1) \quad , \qquad (2.16)$$

$$\left.\begin{aligned} C_1 &= \frac{2}{2+n} \quad , \\ C_2 &= \frac{\gamma-1}{2\gamma+n-2} \quad , \\ C_3 &= \frac{(\gamma-2)\ 2n}{(2+n)(n\gamma-n+2)} - \frac{(2+n)(\gamma-1)\gamma}{(2\gamma+n-2)(n\gamma-n+2)} \quad , \\ C_4 &= \frac{\gamma}{\gamma-2} \quad , \\ C_5 &= \frac{(2\gamma+n-2)2n}{(2+n)(n\gamma-n+2)} + \frac{(2+n)(\gamma-1)\gamma}{(2-\gamma)(n\gamma-n+2)} \quad . \end{aligned}\right\} \qquad (2.17)$$

The constant $K(n,\gamma)$ enters the formulas (2.8), (2.9), and (2.10) through the time functions x_s, u_s, and p_s, respectively. The constant is determined by substituting Eqs. (2.8) through (2.10) into the integral (2.2). After cancelling out the factor E_o one thereby obtains the relation

$$1 = K(n,\gamma)^{n+2} \cdot B \quad ,$$

i. e.

$$K(n,\gamma) = B^{-1/(n+2)} \quad , \tag{2.18}$$

where

$$B = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2})} \cdot \left(\frac{2}{2+n}\right)^2 \cdot \frac{4}{\gamma^2-1} \cdot \int_{v_1}^{v_2} \phi(v)dv \quad , \tag{2.19}$$

$$\phi(v) = \left\{ D_1^{2+(2+n)C_1} D_2/D_4 + 1 \right\} D_2^{nC_2} D_3^{C_5+nC_3} D_4^{C_4} D_5 \quad , \tag{2.20}$$

$$\left.\begin{aligned} D_1 &= \frac{v}{v_2} \quad , \\ D_2 &= \frac{v-v_1}{v_2-v_1} \quad , \\ D_3 &= \frac{1-av}{1-av_2} \quad , \\ D_4 &= \frac{v_1\gamma-v}{v_1\gamma-v_2} \quad , \\ D_5 &= -\frac{C_1}{v} + \frac{C_2}{v-v_1} - \frac{C_3a}{1-av} \quad . \end{aligned}\right\} \tag{2.21}$$

Eqs. (2.8) through (2.21) provide, together with the shock formulas of Section 2.1, explicit expressions for the flow profiles, i.e., formulas for the computation of the flow field for any fixed time t.

2.3. Computation of Flow Histories

Flow history, i.e., the description of the flow at a fixed position, can be obtained from the formulas of Sections 2.1 and 2.2 after a simple manipulation. Let X be the distance from the explosion at which the flow history is to be computed. The shock arrival time at x = X is, according to Eq. (2.1)

$$t_s(X) = K(n,\gamma)^{-(n+2)/2} \left(\frac{\rho_o}{E_o}\right)^{1/2} X^{(n+2)/2} \qquad (2.22)$$

On the other hand, one obtains from Eqs. (2.1) and (2.8) the relation

$$t = K(n,\gamma)^{-(n+2)/2} \left(\frac{\rho_o}{E_o}\right)^{1/2} \left(\frac{X}{y(v)}\right)^{(n+2)/2} \qquad (2.23)$$

Therefore,

$$t = t_s(X)\ y(v)^{-(n+2)/2} \qquad (2.24)$$

If one varies in Eq. (2.24) the parameter v between $v=v_2$ and $v=v_1$, then one obtains time values between $t=t_s(X)$ and infinity, i.e., all times after shock arrival at X. The corresponding values of the other flow variables are obtained by substituting v and the corresponding value of t (computed using Eq. (2.24)) into the Eqs. (2.9), (2.10) and (2.11).

We notice in passing that Eq. (2.8) can be simplified for these calculations by substituting in it the definition of $u_s(t)$ and v_2. The result is

$$u = x(t,v) \cdot \frac{v}{t}\ , \qquad (2.25)$$

where x(t,v) is given by Eq. (2.8). For flow history computations $x(t,v) \equiv X$, of course.

2.4. Particle Path Formulas

Closed formulas for the particle trajectories (or paths) were derived by Laporte and Chang[6,7]. We shall quote their formulas in a somewhat simpler but algebraically equivalent form.

The trajectory of a particle which leaves the shock at the time $t = T$ is given by

$$\left.\begin{aligned} t &= T \cdot w(v) \\ x &= x_s(t) \cdot y(v) \quad , \end{aligned}\right\} \tag{2.26}$$

where

$$w(v) = \frac{v}{v_2} \cdot \left(\frac{\gamma v_1 - v}{\gamma v_1 - v_2}\right)^{P_1} \left(\frac{v-v_1}{v_2-v_1}\right)^{P_2} \left(\frac{1-av}{1-av_2}\right)^{P_3} \quad . \tag{2.27}$$

$$\left.\begin{aligned} P_1 &= (2\gamma+n-2)(2+n)(1-\gamma)\gamma \cdot P_o^{-1} \quad , \\ P_2 &= (2+n)n(\gamma-2)(1-\gamma)\gamma \cdot P_o^{-1} \quad , \\ P_3 &= (2+n)^2(\gamma-1)^2\gamma \cdot P_o^{-1} - 1 \quad , \end{aligned}\right\} \tag{2.28}$$

$$\begin{aligned} P_o = 2 \ &(2n\gamma + 2\gamma-n + 2)(2-n-2\gamma) + 2(n\gamma+4)n(2-\gamma)\gamma + \\ &+2(n\gamma-n+2)(2+n)(\gamma^2-1) \quad . \end{aligned} \tag{2.29}$$

The constants v_1, v_2, and a are defined by Eqs. (2.15) and (2.16), respectively. The function y(v) is defined by Eq. (2.12), and $x_s(t)$ is defined by Eq. (2.1).

In Eq. (2.26), one obtains with $v=v_2$ the initial point of the particle path, i.e., a point on the shock and with the coordinates t=T and $x=x_s(T)$. By decreasing the parameter v to $v=v_1$ one obtains t and x values that approach infinity, thus covering the complete particle trajectory.

The flow variables u, p and ρ are obtained at any point of the path line by substituting the corresponding values of v and t into the Eqs. (2.9), (2.10) and (2.11), respectively.

2.5. Mach-Line Formulas

Closed form expressions for the Mach-lines in a strong blast field were derived by Laporte and Chang[6,7]. We shall present their formulas in a somewhat simpler but algebraically equivalent form.

The Mach-lines are given in terms of the dimensionless parameter v by

$$\left.\begin{aligned} t &= M_{\pm}\cdot t_1(v)\cdot e^{\pm t_2(v)} \quad , \\ x &= x_s(t)\cdot\, y(v) \quad , \end{aligned}\right\} \tag{2.30}$$

where

$$\left.\begin{aligned} t_1(v) &= \frac{av}{1-av} \quad , \\ t_2(v) &= \frac{1}{\sqrt{b_1+b_2}}\ \text{arc sin}\ \frac{b_1+b_2 av_2}{av_1(1-av)} \quad , \end{aligned}\right\} \tag{2.31}$$

$$\left.\begin{aligned} b_1 &= \Big[\, 2\gamma av_1 - (\gamma+1)\Big]\frac{av_1}{\gamma-1} \quad , \\ b_2 &= \Big[\, 2 - (\gamma+1)av_1\Big]\frac{1}{\gamma-1} \quad . \end{aligned}\right\} \tag{2.32}$$

The function y(v) is defined by Eq. (2.12), $x_s(t)$ is defined by Eq. (2.1), and the constants v_1, v_2 and a are defined by Eqs. (2.15) and (2.16), respectively.

The constants M_+ and M_- in Eq. (2.30) correspond to the positive and negative sign of the exponent function $t_2(v)$, respectively, and determine the two Mach-lines passing through each point in the x,t-plane.

In order to calculate a specific characteristic one might specify as initial point either a point on the shock or a point with x=0 (i.e., at the location of the explosion).

Let the initial point be on the shock at a distance X from the explosion. Then the corresponding shock arrival time $T=t_s(X)$ can be computed with Eq. (2.22). The two constants M_+ and M_- which define the two characteristics crossing at (X,T) are obtained by substituting $v=v_2$ and t=T into Eq. (2.30). The result is

$$\left.\begin{aligned} M_+ &= \frac{T}{t_1(v_2)} \quad e^{-t_2(v_2)} \\[2ex] M_- &= \frac{T}{t_1(v_2)} \quad e^{+t_2(v_2)} \end{aligned}\right\} \qquad (2.33)$$

If the starting point for the calculation of the characteristics is chosen at the explosion (x=0) and time t=T, then the constants M_+ and M_- are obtained from Eq. (2.30) for $v=v_1$. Substituting $v=v_1$ into Eq. (2.31) one obtains

$$t_2(v_1) = - \frac{\pi}{2} / \sqrt{b_1+b_2} \qquad (2.34)$$

Therefore, in this case the formulas for the constants simplify to

$$\begin{aligned} M_+ &= \frac{T}{t_1(v_1)} \quad \exp\left(\frac{\pi}{2} \; / \sqrt{b_1+b_2}\right) \\[2ex] M_- &= \frac{T}{t_1(v_1)} \quad \exp\left(- \frac{\pi}{2} \; / \sqrt{b_1+b_2}\right) \end{aligned} \qquad (2.35)$$

Once the constants M_+ and M_- are found, then all other points of each characteristic are obtained by letting the parameter v vary between v_1 and v_2. The former value corresponds to a point with x=0 and the latter value corresponds to a point on the shock.

The flow variables u, p, and ρ along the Mach-lines are obtained by substituting the corresponding values of v and t into the Eqs. (2.9), (2.10), and (2.11), respectively.

3. COMPUTER PROGRAMS

3.1. General Comments

The computer program package consists of six subroutines for flow calculations and two auxiliary routines. One of the subroutines, SBLPREP, is a preparation routine. It computes a set of constants from input data (ambient conditions and energy released) and makes the constants available to the other programs via a labeled COMMON. The preparation routine must be called first, before any of the other flow calculation routines can be called.

The other five flow calculation routines compute, respectively, the shock, the flow profile at a given time, the flow history at a given distance, a particle path with corresponding flow variables, and a pair of Mach-lines with corresponding flow variables.

The two auxiliary routines are used by the preparation routine for its calculation of the constants.

All arguments are assumed to be expressed in SI base units.

3.2. Preparation Routine SBLPREP

The preparation routine SBLPREP computes the values of all constants that are needed for flow calculations and appear in the formulas of Section 2. The subroutine must be called (once) before any of the actual flow calculation routines can be used. The preparation routine is called by the statement

```
Call SBLPREP(N,P,TEM,GAM,AMOL,ENCHRG,NBAD)
```

The arguments are

N = dimension of the space (N=1,2, or 3);

P = ambient pressure (in pascals);

TEM = ambient temperature (in kelvins);

GAM = ratio of specific heats; GAM is restricted by Eq. (2.7);

AMOL = molar mass of the ambient gas (in kg/mole);

ENCHRG = energy released by the charge (in $J \cdot m^{N-3}$; one kiloton TNT equivalent is $4.184 \cdot 10^{12}$J, *).

NBAD = an error indicator; it is set equal to zero by SBLPREP after computation of all constants; a non-zero NBAD on return indicates that the constants cannot be computed with the values given by the arguments N through ENCHRG.

**The equivalence 1 kton TNT = $4.184 \cdot 10^{12}$J is given in reference 13. In reference 14, p. 13, footnote, a kton TNT is defined as 10^{12} calories without specifying the calorie type, and in the same reference, pp. 13 and 647 one finds the conversion factors 4.18, 4.187, and $4.2 \cdot 10^{12}$. According to reference 13, a calorie has a value between 4.1819J and 4.19002 J, depending on its type. In the present context, the appropriate type seems to be a thermochemical calorie" which is defined as 4.184J.*

[13] *American National Standards Institute, American Society for Testing Materials, "Standard for Metric Practice", ASTM E380-76, ASTM, Philadelphia, PA, 1976.*

[14] *Samuel Glasstone and Philip F. Dolan, eds, "The Effects of Nuclear Weapons", 3rd edition, U.S. Department of Defense and U.S. Department of Energy, 1977.*

If any of the quantities P through AMOL are missing (i.e., are not positive), then the following default values are used instead of the missing quantities:

P = 101.325 kPa

TEM = 293.0 K

GAM = 1.4

AMOL = 28.96 g/mole

These values correspond to a "standard" air. The use of default values is indicated by a printed message.

If the charge energy ENCHRG is not positive, or if N or GAM are not within the indicated ranges, then the calculation of the constants is not carried out. Instead, a return with NBAD $\neq$ 0 is executed, after printing of a corresponding error message. If GAM = 2.0 is specified by the calling routine, then the computation will be carried out with the default value GAM = 1.99.

The calculated constants are stored in a labeled COMMON/SBLCOM/ and made thus available to the flow calculation routines. The contents of SBLCOM is printed by the subroutine before executing a regular return.

The calculation of all but one of the constants is trivial and involves merely the evaluation of the lengthy formulas of Section 2. The computation of the constant $K(n,\gamma)$, defined by Eqs. (2.18) through (2.21), requires a numerical quadrature, which is carried out using a Romberg algorithm. The algorithm is implemented by the auxiliary routines SBLROMB and SBLINTE. Because the integrand $\phi(v)$ is singular at the lower limit v_1, the integration is done in terms of the transformed variable

$$X(v) = \left(\frac{v - v_1}{v_2 - v_1}\right)^{1/(nC_2)} ,$$

between the limits zero and one.

The input arguments do not contain the density of the ambient air, which is needed in the formulas. The density is calculated in SBLPREP from the given arguments by the formula.

$$\rho_o = (P*AMOL)/(TEM*8.3143) ,$$

where 8.3143 is the universal gas constant in J/(K·mole).

The pressure and temperature instead of density were chosen as input parameters because the former quantities are more readily available when dealing with simulations of a real explosion.

3.3. Shock Computation Using SBLSHCK

The subroutine SBLSHCK computes the trajectory of a shock generated by a strong blast, and corresponding shock and particle velocities, shock pressure, shock density, and dynamic pressure. The computations are done for a prescribed number of distances from the explosion, equidistantly spaced between X_{min} and X_{max}.

The subroutine is called by the statement

```
CALL SBLSHCK (XMIN, XMAX, NRX, X, T, USHCK, P, UPART, RHO, DYNPR, NBAD)
```

The first three arguments are to be specified by the calling program. They are

XMIN, XMAX = limits of distances (in metres) from the explosion between which calculations should be done.

NRX = number of nodes to be calculated between XMIN and XMAX.

The next eight arguments (X through DYNPR) are one-dimensional arrays in which the calculated results will be stored:

X = array of distances of the nodes in metres,

T = array of shock arrival times at the nodes in seconds,

USHCK = array of shock velocities (=dx/dt) in m/s,

P = array of shock pressures in pascals (P may be also interpreted as overpressure, because the ambient pressure is assumed to be negligible);

UPART = array of particle velocities at the shock in m/s,

RHO = array of densities at the shock in kg/m^3,

DYNPR = 0.5*RHO * UPART ** 2 = array of dynamic pressures in pascals.

The last argument, NBAD, is an error indicator. It is set equal to zero if all computations have been properly carried out. If NBAD is not zero at return, then the shock has not been calculated. A printed error message explains the reason for such an error return. It can be caused, e.g., by an NRX <0, by XMIN>XMAX, etc.

The formulas for the calculation of the shock are given in Section 2.1.

3.4. Flow Profile Computation Using SBLPROF

The subroutine SBLPROF computes flow profiles, i.e., the flow field for fixed times. The calculations are based on formulas for the flow field of a strong blast. The formulas are given in Section 2.2.

The subroutine is called by the statement

```
CALL SBLPROF(T,XMIN,XMAX,NRX,X,P,UPART,RHO,DYNPR,NBAD)
```

The subroutine computes for the time T a total of NRX nodes located at distances between XMIN and XMAX from the explosion. The first four arguments, T through NRX, are set by the calling program. They are:

T = time (in seconds) after the explosion, for which the profile should be computed;

XMIN, XMAX = range of distances (in metres) from the explosion, between which the profile should be computed (the actual computation will be done only for positive distances and only within the blast region);

NRX = number of nodes to be computed between XMIN and XMAX.

The remaining arguments are set by the subroutine SBLPROF. The arguments X through DYNPR are one-dimensional arrays in which the subroutine will store the coordinates of the calculated nodes and the corresponding values of flow variables. The arguments are:

X = x-coordinates of the nodes (distances from the explosion in metres);

P = pressures in pascals;

UPART = particle velocities in m/s;

RHO = densities in kg/m^3;

DYNPR = dynamic pressures in pascals

= 0.5* RHO * UPART ** 2

The last argument, NBAD, is an error indicator. It is set equal to zero by the subroutine if the profile has been calculated. If the profile cannot be calculated for the specified arguments, then NBAD assumes a non-zero value. In such a case also an error message is printed explaining the reasons for the error return.

The computed nodes are not equidistant in terms of X. Instead, the distances between the nodes are smaller in the vicinity of the shock. Other arrangements of nodes can be obtained by calling the subroutine repeatedly, e.g., once for every two nodes, specifying NRX=2 and assigning corresponding values for XMIN and XMAX.

3.5. Flow History Computation Using SBLHIST

The subroutine SBLHIST computes flow histories at prescribed distances from the explosion. The computations are based on formulas given in Section 2.3. The subroutine is called by the statement

```
CALL SBLHIST(X,TMIN,TMAX,NRT,T,P,UPART,RHO,DYNPR,NBAD)
```

The subroutine computes NRT nodes located at the distance X from the explosion and between the times TMIN and TMAX after the explosion. The first four arguments, X through NRT, are set by the calling program. They are:

X = distance (in metres) from the explosion at which the history should be computed;

TMIN,TMAX = minimum and maximum times (in seconds) after the explosion for which the computations should be done; the first node will be calculated for t = max (TMIN,t_s), where t_s is the shock arrival time at the distance X (Eq. (2.22));

NRT = number of nodes to be computed.

The remaining arguments are set by the subroutine. The arguments T through DYNPR are one-dimensional arrays in which the subroutine stores the computed nodal values. The arguments are:

T = array of times (in seconds) after the explosion;

P = array of pressures (in pascals);

UPART = array of particle velocities (in m/s);

RHO = array of densities (in kg/m^3);

DYNPR = array of dynamic pressures (in pascals) = 0.5* RHO*UPART** 2

The last argument, NBAD, is an error indicator. It is set equal to zero by the subroutine after the computations have been completed. In case the flow history cannot be computed for the given arguments, NBAD is assigned a non-zero value and a message is printed explaining the reason for the error return.

The computed nodes are approximately equidistant in time, but the distances between the nodes are smaller in the vicinity of the shock. Other node arrangements can be obtained by calling the subroutine repeatedly with proper values of TMIN and TMAX and with specifying NRT=1 or NRT=2.

3.6. Particle Path Computation Using SBLPATH

The subroutine SBLPATH computes particle trajectories in a strong blast flow field. The initial point of the trajectory is a point on the shock. The end point is specified by a maximum time or a maximum distance. The computations are based on formulas given in Section 2.4.

The subroutine is called by the statement

```
CALL SBLPATH(XSHCK,TMAX,XMAX,NRNOD,X,T,P,UPART,RHO,DYNPR,NBAD)
```

The first four arguments are set by the calling routine. They are:

XSHCK = distance (in metres) of the initial point of the trajectory from the explosion, i.e., the x-coordinate of a shock point;

TMAX, XMAX = coordinates of the end point of the trajectory; the path will end either at time = TMAX, or at distance = XMAX, whichever is reached first;

NRNOD = number of nodes to be computed.

The remaining arguments are set by the subroutine. The arguments X through DYNPR are one-dimensional arrays in which the subroutine stores the coordinates and flow variable values of the NRNOD computed nodes. The arguments are

X = distances (in metres) of nodes from the explosion;

T = times (in seconds) after explosion;

P = pressures (in pascals);

UPART = particle velocities = dx/dt, (in m/s);

RHO = densities (in kg/m^3);

DYNPR = dynamic pressures (in pascals) = 0.5* RHO*UPART** 2.

The last argument, NBAD, is an error indicator. It is set equal to zero by the subroutine if the particle path has been computed. If the path cannot be computed with the data specified by the calling program then NBAD is assigned a non-zero value and a message is printed by SBLPATH explaining the reasons for the error return.

The nodes are approximately equidistant in time.

3.7. Mach-Line Computation Using SBLMACH

The subroutine SBLMACH computes Mach-lines in a flow field of a strong blast. The user has to specify an initial point of the Mach-lines either on the shock, or at the site of the explosion. The subroutine then establishes both Mach-lines that pass through the specified point. The actually computed nodes are located on segments of these characteristics, defined by minimum and maximum values of distance and time. The computation is based on the formulas of Section 2.5.

The subroutine is called by the statement

```
CALL SBLMACH(XZ,TZ,XMIN,XMAX,TMIN,TMAX,NRNODE,X,T,P,UPART,RHO,DYNPR,NBAD)
```

The first seven arguments, XZ through NRNODE, are set by the calling program. They are:

XZ = distance (in metres) of the initial point from the explosion (XZ must be zero if TZ is positive and vice versa);

TZ = time (in seconds) after explosion for the initial point; if TX>0 and XZ = 0 then the initial point is assumed to be at the site of the explosion; if TX = 0 and XZ > 0 then the initial point is assumed to be on the shock at the distance XZ from the explosion;

XMIN,XMAX
TMIN,TMAX = limits between which the Mach-lines should be calculated i.e., limits of a computational "window".

NRNODE = array containing the numbers of nodes to be computed; the array contains two values, NRNODE(1) is the number of nodes for the characteristic with dx/dt > 0, and NRNODE(2) is the number of nodes for the other characteristic.

The remaining seven arguments are set by the subroutine. The arguments X through DYNPR are two-dimensional arrays in which the subroutine stores the nodal values of the characteristics. The dimensions of the arrays are (2,NMAX), where NMAX is the maximum number of nodes to be computed i.e., NMAX must be larger than or equal to max(NRNODE(1), NRNODE(2)). The nodes with the index (1, K) belong to the characteristic with dx/dt > 0, and the nodes with the index (2, K) belong to the other characteristic.

For clarity, we show the arguments with their dimensions. The arguments are:

X(2,NMAX) = distances (in metres) of the nodes from the explosion;

T(2,NMAX) = times (in seconds) after the explosion;

P(2,NMAX) = pressures (in pascals);

UPART(2,NMAX) = particle velocities (in m/s);

RHO(2,NMAX) = densities (in kg/m^3);

DYNPAR(2,NMAX) = dynamic pressures (in pascals) = 0.5* RHO*UPART**2

The argument NBAD is an error indicator. It is set equal to zero by the subroutine after the Mach-lines have been computed. If they cannot be computed for the specified input arguments, then NBAD is assigned a non-zero value and an error message is printed by SBLMACH explaining the reason for the error return.

If only one Mach-line should be computed, then the user may specify NRNODE = 0 for the other Mach-line.

4. EXAMPLE

As an example we present calculations of a one megaton TNT explosion in air assuming spherical symmetry. The calculations were started by calling the preparation subroutine with the statement

```
CALL SBLPREP (3,0.,0.,0.,0.,4.184 E15,NBAD).
```

The first argument advises the program that the explosion is spherically symmetric ("three-dimensional"). The next four arguments describe the ambient conditions. Because they are not positive the subroutine SBLPREP assigns standard air default values for the ambient pressure, temperature, ratio of specific heats and molar mass (see Section 3.2). The charge energy is specified as $4.184 \cdot 10^{15}$J, which equals one megaton TNT equivalent, expressed in joules. (The conversion factor is $4.184 \cdot 10^{9}$J for one ton TNT equivalent[13].)

The calculation results are shown in Figures 1 through 6. Figures 1 and 2 show flow profiles at 0.4 seconds after the explosion. The profiles were obtained by calling the subroutine SBLPROF. Figures 3 and 4 present flow histories at 800 m from the explosion. They were obtained by plotting the results from the subroutine SBLHIST. Figure 5

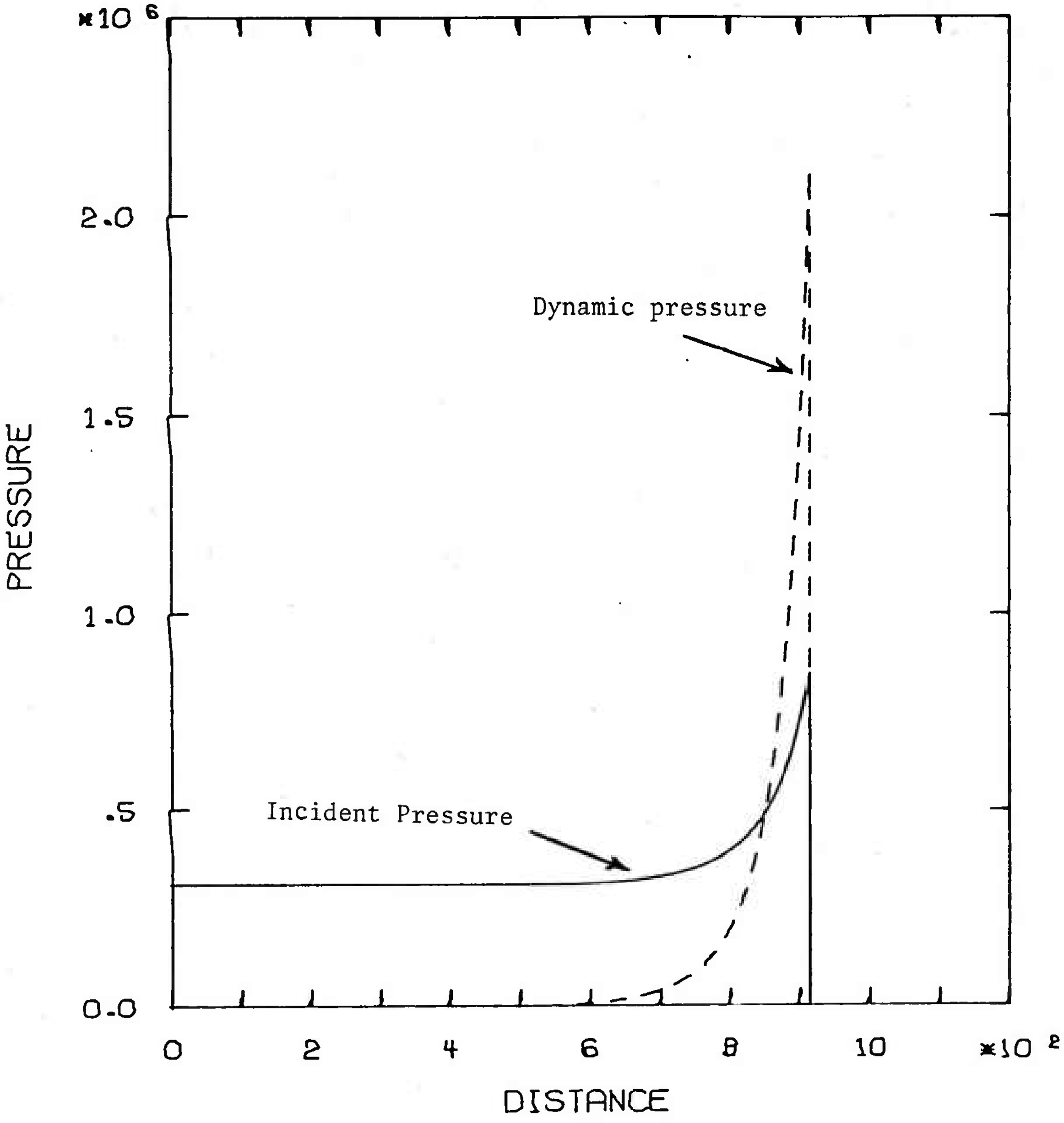

Figure 1. Pressure Profiles 0.4 Seconds After the Explosion.

Distances are expressed in metres and pressures are expressed in pascals. Dynamic pressure is defined as the product 0.5 · ρu^2.

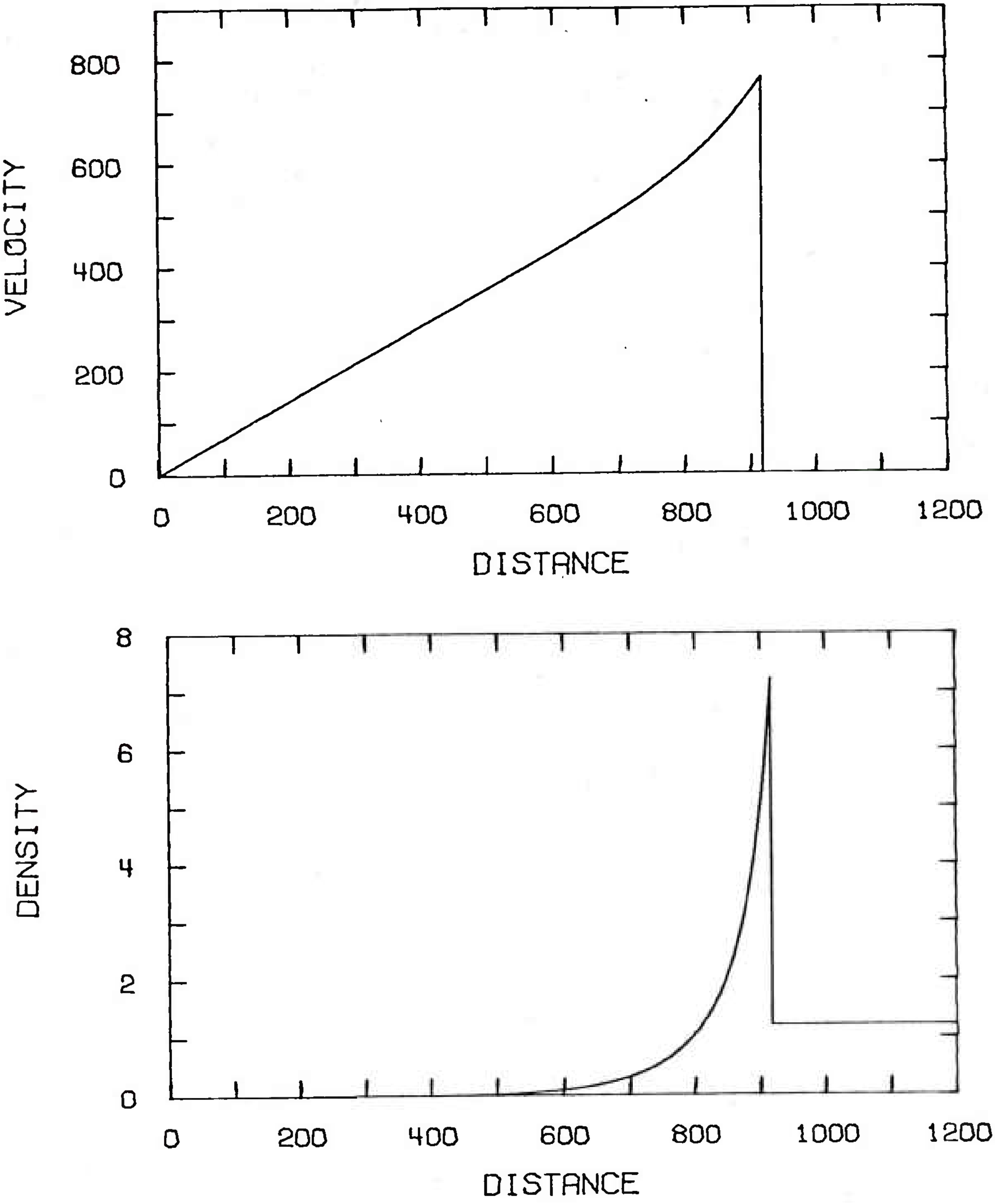

Figure 2. Velocity and Density Profiles 0.4 Seconds After the Explosion.

Velocities are expressed in m/s, densities are expressed in kg/m^3, and distances are expressed in metres.

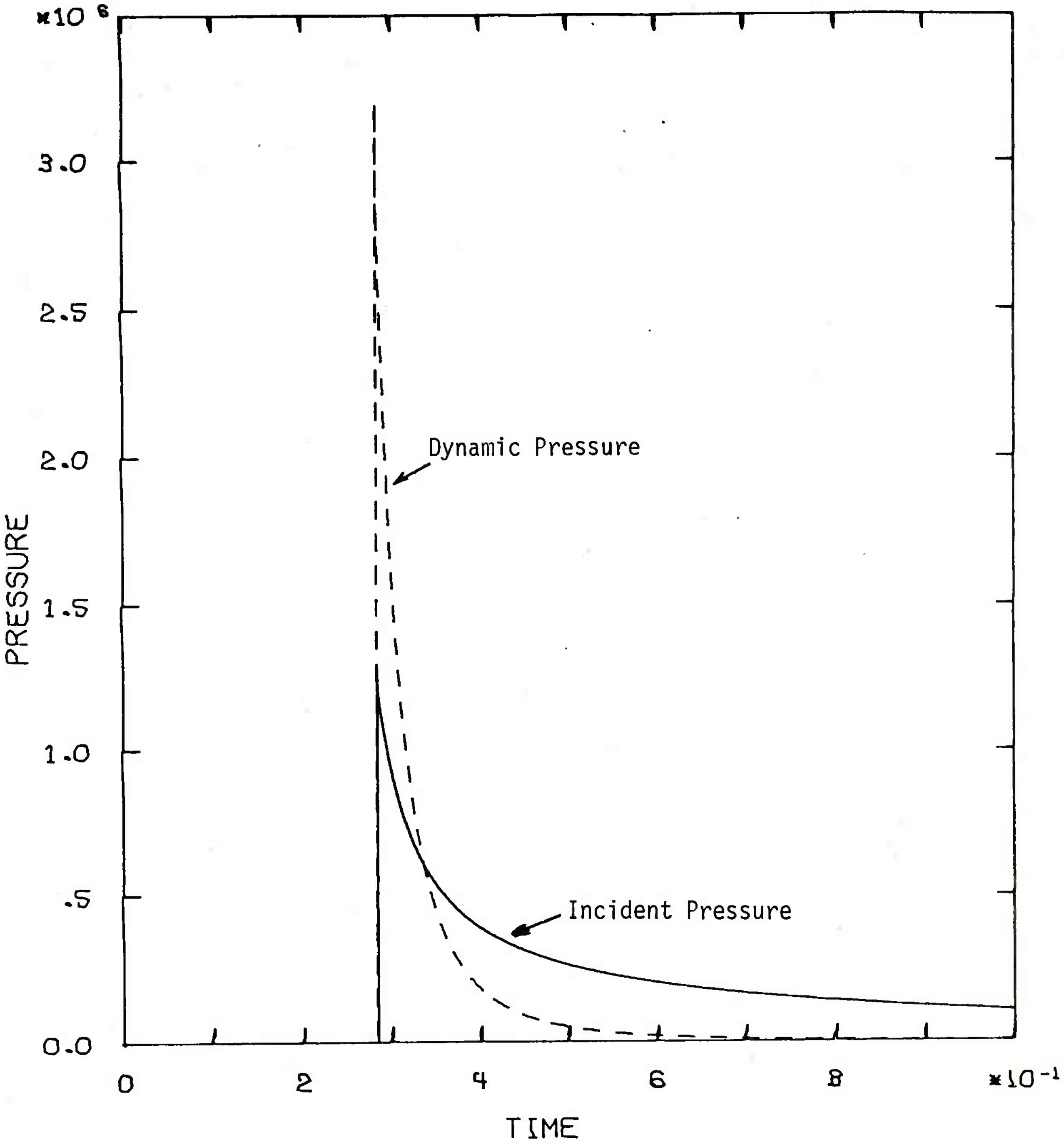

Figure 3. Pressure Histories at 800 m from the Explosion.

Times are expressed in seconds and pressures are expressed in pascals. Dynamic pressure is defined as the product 0.5 · ρu^2.

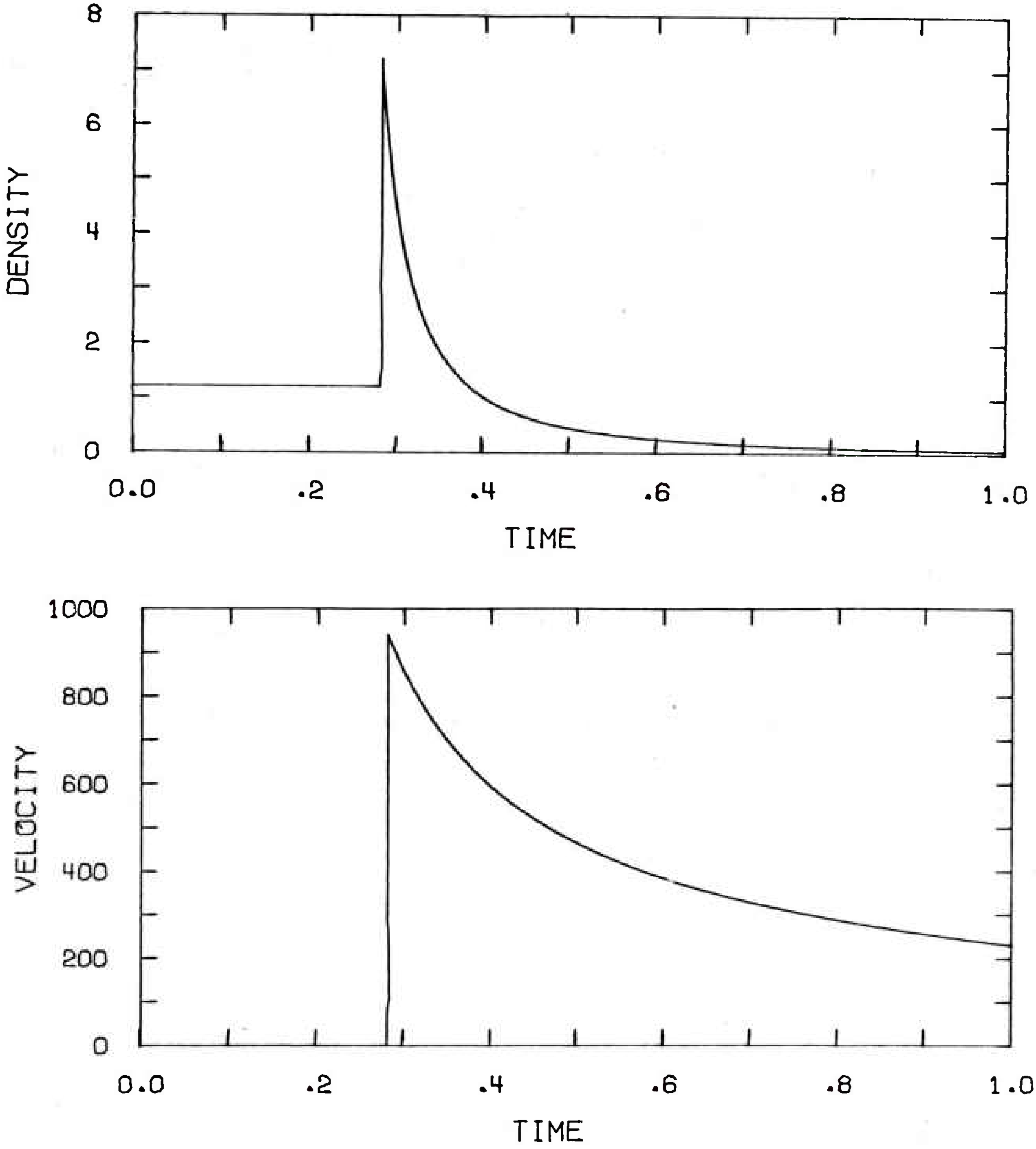

Figure 4. Velocity and Density Histories at 800 m from the Explosion.

Velocities are expressed in m/s, densities are expressed in kg/m^3 and times are expressed in seconds.

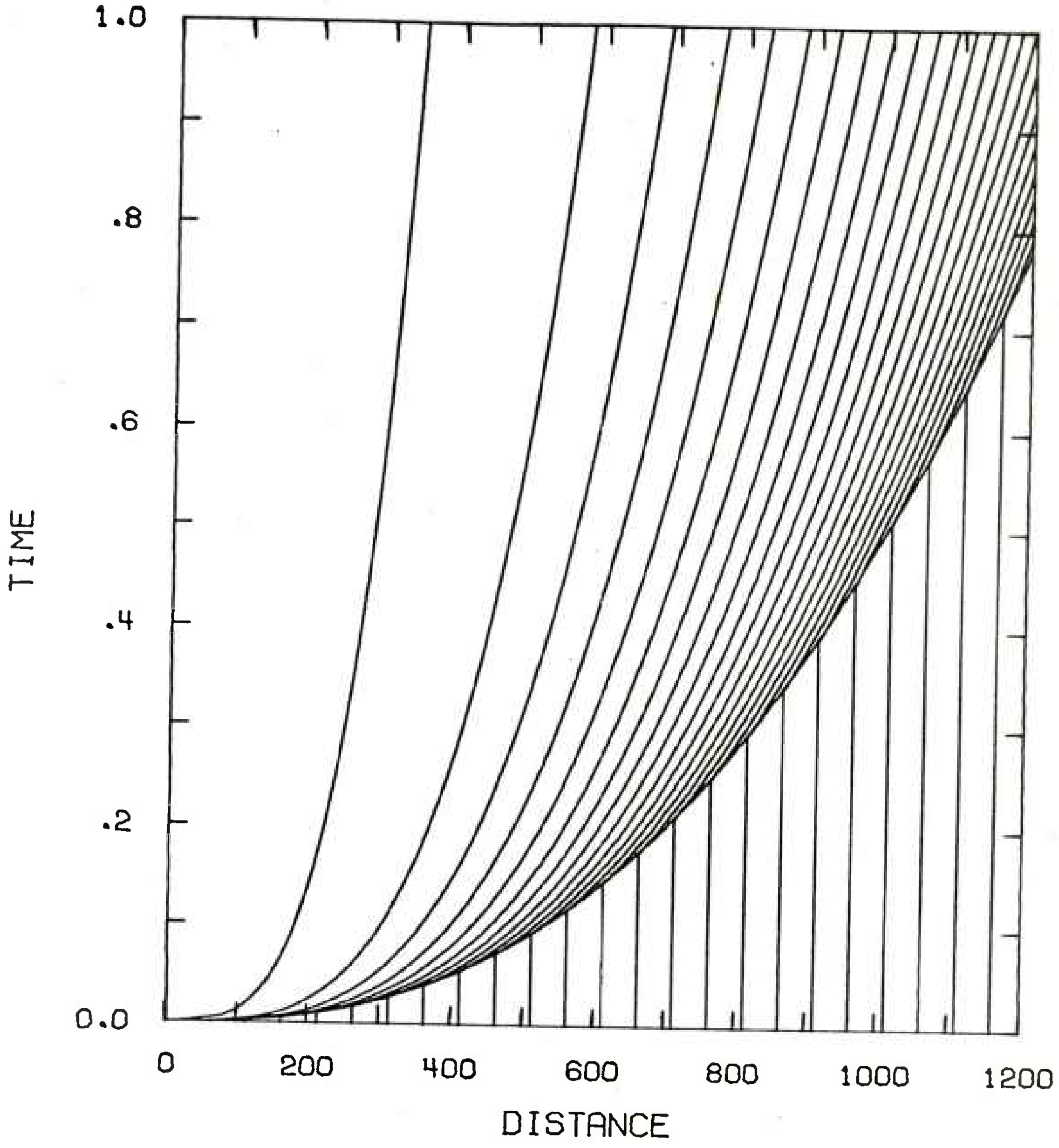

Figure 5. Particle Trajectories of a one megaton TNT Explosion. Distance is expressed in metres and time is expressed in seconds.

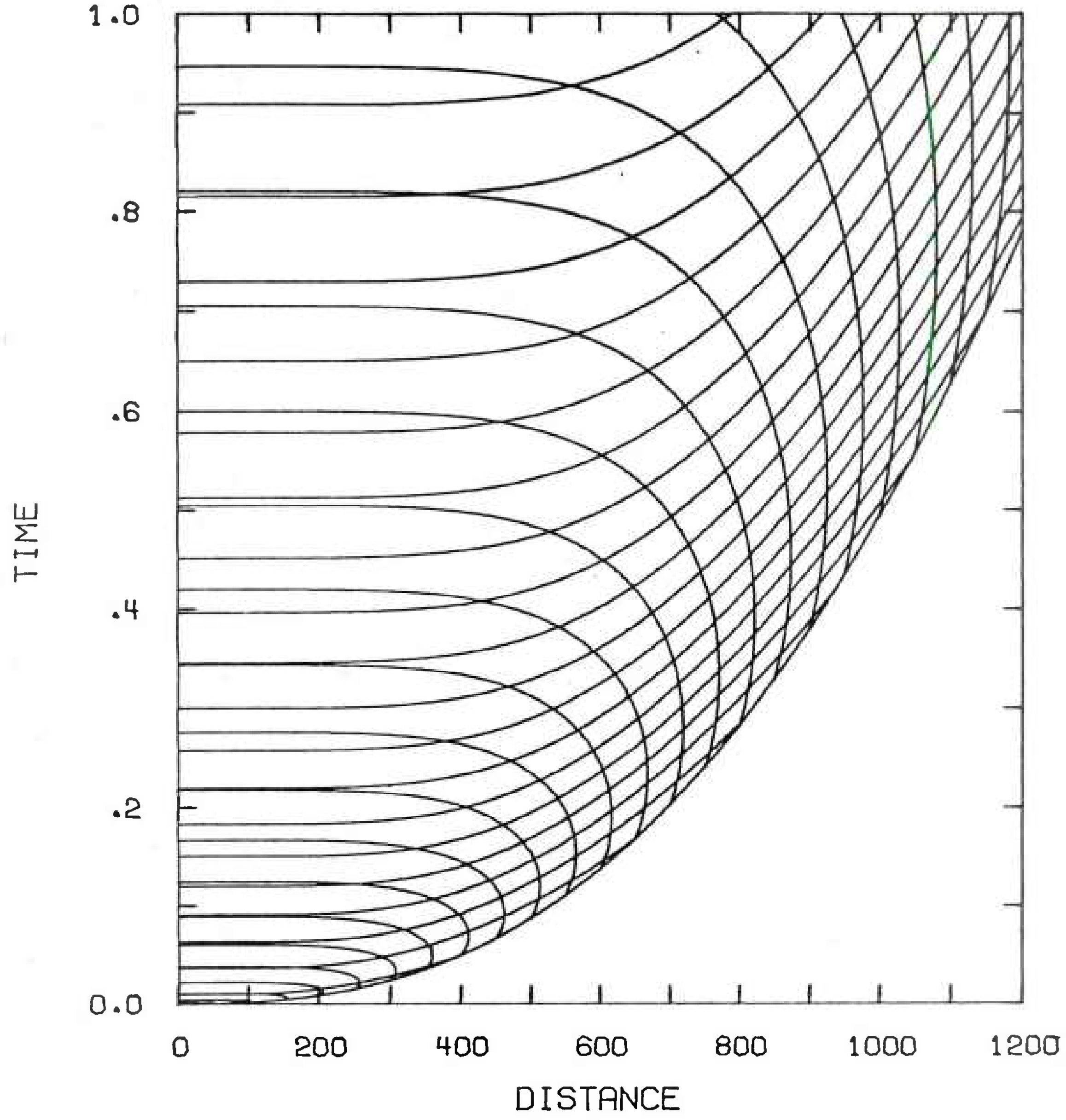

Figure 6. Mach-lines in the Blast Bubble of a one megaton TNT Explosion.

Distance is expressed in metres and time is expressed in seconds.

shows some particle paths computed using the subroutine SBLPATH, and Figure 6 shows Mach-lines computed by the subroutine SBLMACH. The shock line in Figure 5 and Figure 6 was obtained by calling the subroutine SBLSHCK.

5. LIMITS OF POINT SOURCE APPROXIMATIONS OF EXPLOSIONS IN AIR

The strong blast formulas of Section 2 are based on assumptions that are outlined in Section 1 and that might be satisfied at distances not too close nor too far from the explosion. In order to make the formulas useful as approximations to real explosions, one has to specify more precisely their range of applicability. In this section we shall derive quantitative conditons for which the strong blast formulas approximate spherical explosions in air. The derivations will be based on calculations in references 15 and 16. Both references report results obtained by numerical solution of so-called bursting sphere problems.

A bursting sphere problem is characterized as follows: At time $t = 0$ one postulates an ideal gas in which an amount of internal energy E is uniformly distributed throughout a sphere of radius R. The sphere can be thought of as representing the fireball formed during the initial stages of an explosion when thermal and chemical processes dominate. The time zero is the time at which the fireball is completely formed and the hydrodynamic motion starts by forming a shock, which subsequently propagates into the ambient atmosphere. The initial velocities are assumed to be zero everywhere, and the initial density is assumed to be constant and equal ρ_1 within the sphere, and constant and equal ρ_o outside the sphere. (In reference 15 it is also assumed that $\rho_1 = \rho_o$.) Let the corresponding initial pressures be p_1 and p_o , respectively.

First we discuss the dependence of the incident shock pressure on distance, as obtained from strong blast formulas and flow field calculations, respectively. The results from bursting sphere calculations in reference 15 are displayed in Figures 7 and 8. The curves in the figures represent shock pressures in an ideal gas with the ratio of specific heats $\gamma = 1.4$. The ordinate in Figure 7 is the ratio p_s/p_{so}, i.e., the ratio of shock pressure to initial shock pressure (at time zero). The abscissa is the ratio of distance r to the initial sphere radius R. One has the following relations between the initial shock pressure p_{so}, the initial pressure p_1 in the sphere and the energy E:

[15] *M. Lutzky and D. Lehto, "Transformations for Scaling of Close-In Pressures from Nuclear Explosions", Naval Ordnance Laboratory Report NOLTR-66-12, Silver Spring, MD, March 1966.*

[16] *Shih Lien Huang and Pei Chi Chou, "Calculations of Expanding Shock Waves and Late-Stage Equivalence", Drexel Institute of Technology Report 125-12, Philadelphia, PA, April 1968.*

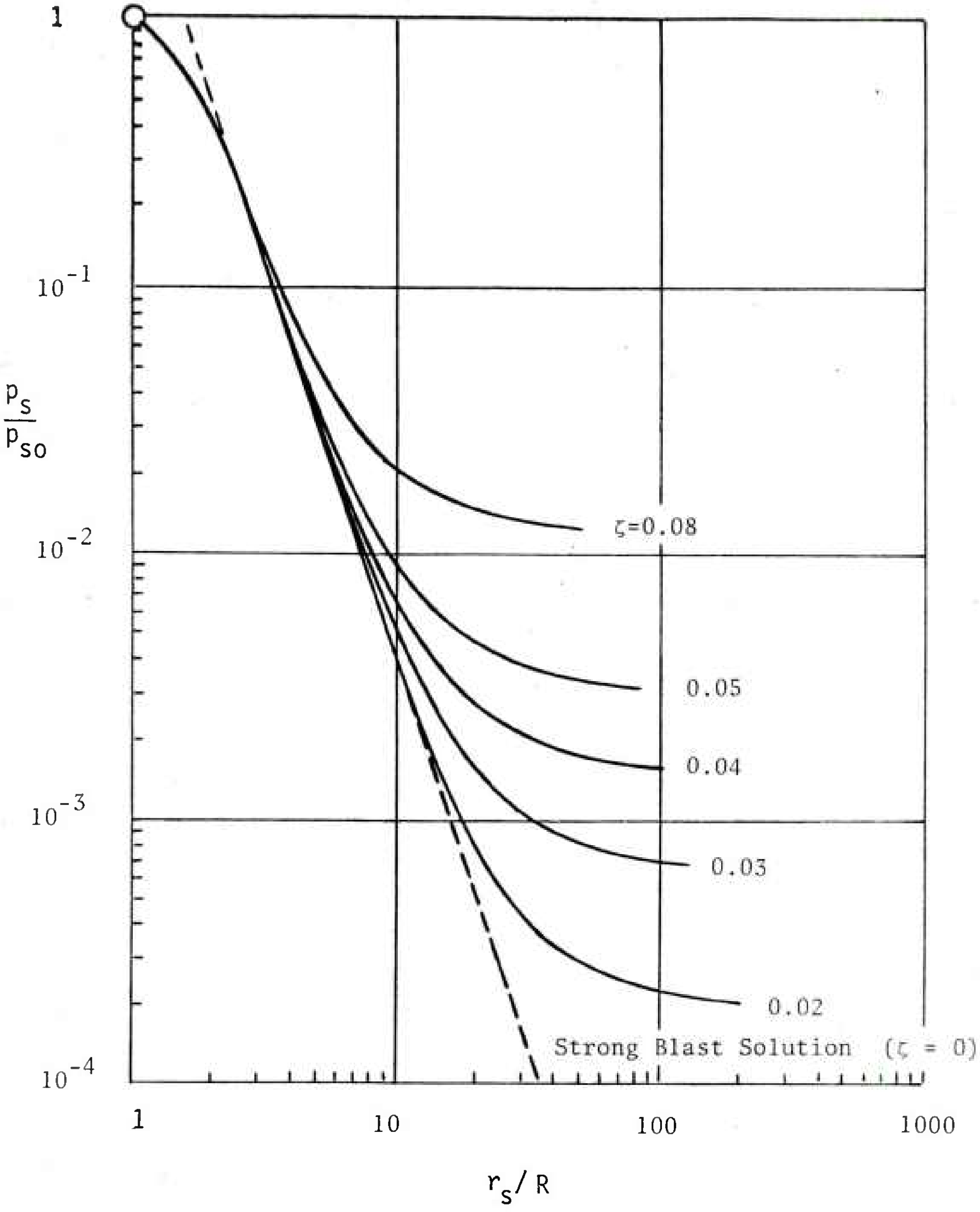

Figure 7. Shock Pressure versus Distance for Bursting Spheres. The curves are taken from reference 15.

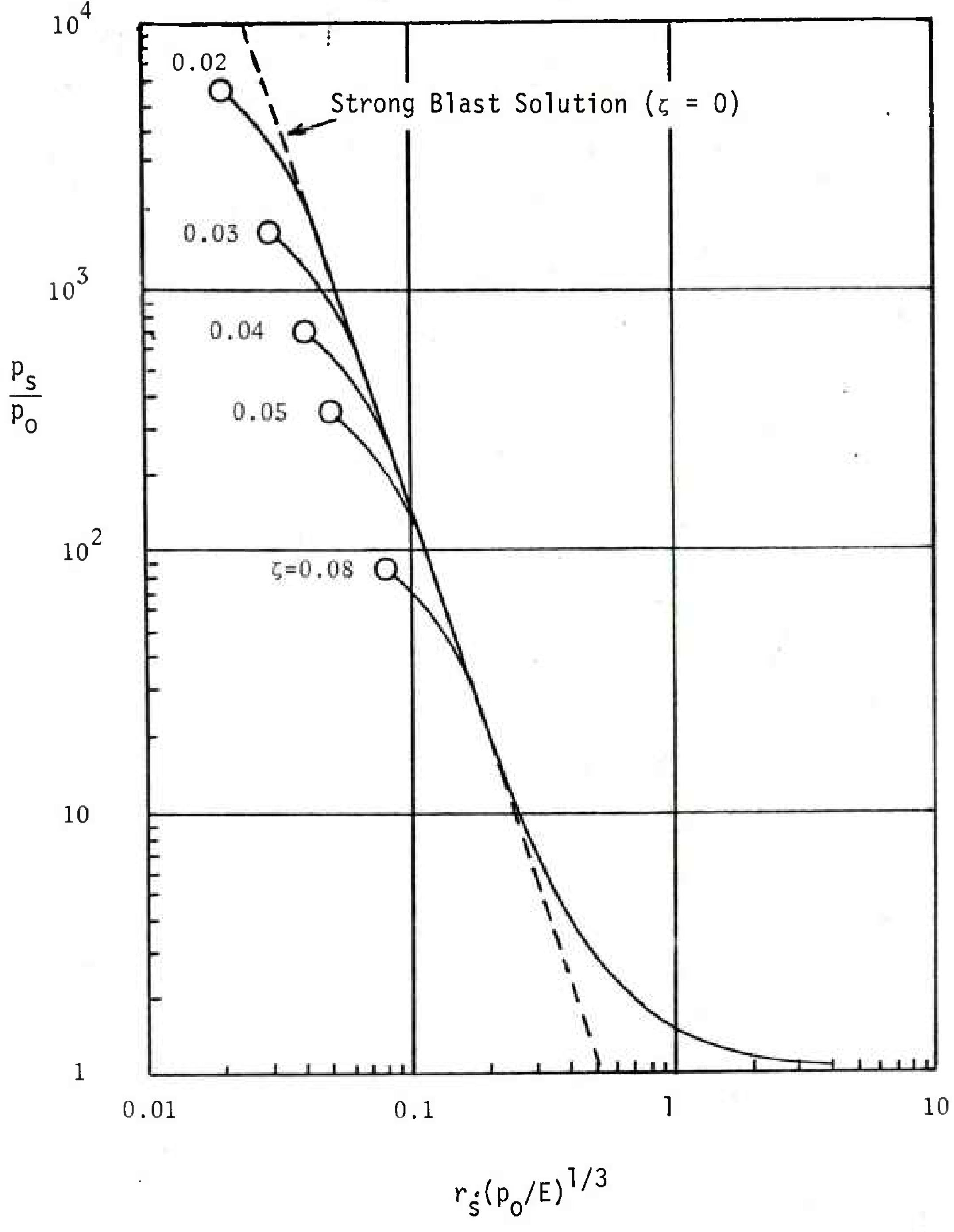

Figure 8. Shock Pressure versus Distance for Bursting Spheres (Sachs' Scales).

The figure is taken from reference 15.

$$p_{so} = 0.461 \cdot p_1 = 0.0440 \cdot E/R^3 . \qquad (5.1)$$

(These and other relations in this section are derived in reference 15 for $\gamma=1.4$ and $p_1/p_o >> 1$.) The strength of the explosion is characterized by the dimensionless parameter

$$\zeta = \sqrt[3]{\frac{R^3 p_o}{E}} = 0.453 \sqrt[3]{\frac{p_o}{p_1}} . \qquad (5.2)$$

The dashed line in Figure 7 represents the strong blast solution. According to Section 2.1 that solution is

$$p_s = \frac{2}{\gamma+1} \cdot \left(\frac{2}{2+n}\right)^2 \cdot K(n,\gamma)^{2+n} \cdot E / x_s^n \qquad (5.3)$$

In the present case, $n = 3$, $\gamma=1.4$, $K(3, 1.4) = 1.03278$ and

$$p_s = 0.1567 \cdot E/r_s^3 , \qquad (5.4)$$

or, by substituting Eq. (5.1) into Eq. (5.4),

$$\frac{p_s}{p_{so}} = 3.56 \cdot (R/r_s)^3 \qquad (5.5)$$

It is apparent from Figure 7 that the strong blast solution approximates a finite segment of the shock curve only for events with $\zeta < 0.08$. In terms of initial pressures or energy, this condition can be formulated by either of the following equivalent relations

$$\left.\begin{aligned} p_{so}/p_o &> 86 , \\ p_1/p_o &> 190 , \\ E/(R^3 p_o) &> 1950 . \end{aligned}\right\} \qquad (5.6)$$

If the event satisfies the conditions (5.6) then the strong blast solution can be used as an approximation between $r/R = 2.5$ and an upper limit which depends on the strength of the explosion.

The upper limit for the approximation can be obtained from Figure 8 which contains the same curves as Figure 7, but uses Sachs' scaling for its coordinates. Figure 8 shows that an upper limit is given by the condition $p_s/p_o > 10$.

Both conditions can be expressed either in terms of distances or in terms of pressures. In terms of distances one obtains the following interval in which the strong blast solution can be used as an approximation:

$$2.5 < \frac{r}{R} < \left\{ \begin{array}{l} 0.25/\zeta \\ 0.25 \sqrt[3]{E/(R^3 p_o)} \\ 0.547 \sqrt[3]{p_1/p_o} \\ 0.707 \sqrt[3]{p_{so}/p_o} \end{array} \right. \qquad (5.7)$$

In terms of pressures one has the equivalent conditions

$$\left. \begin{array}{l} 0.0101/\zeta^3 \\ 0.0101 \cdot E/(R^3 p_o) \\ 0.106 \cdot p_1/p_o \\ 0.230 \cdot p_{so}/p_o \end{array} \right\} > \frac{p_s}{p_o} > 10. \qquad (5.8)$$

In order to check either of the conditions one needs in addition to the energy E also an estimate of the fireball radius R. Such an estimate is often not available. It is, therefore, desirable to have conditions which are independent of a. If one is content with more restrictive conditions than (5.7) or (5.8), then one can replace p_{so} in the last Eq. (5.8) by an observed maximum p_s value. Let that value be p_{smax}. Then a condition for the applicability of strong blast shock formulas to explosions in air is the following set:

$$\left.\begin{aligned} &\frac{p_s}{p_{smax}} < 0.23 \ , \\ &\text{and} \\ &\frac{p_s}{p_o} > 10 \ . \end{aligned}\right\} \tag{5.9}$$

An observation of P_{smax} can be also used to obtain an upper limit for the fireball radius. That limit is, according to Eq. (5.1) given by

$$R < 0.353 \sqrt[3]{E/p_{smax}} \ . \tag{5.10}$$

The flow field generated by a bursting sphere contains in addition to the leading shock also a second shock and a contact discontinuity. Therefore, the strong blast formulas may approximate the field only within a strip in the r,t-plane behind the leading shock, where the flow is free from these singularities. In order to establish limits for approximations within the strip one needs examples of flows generated by explosions for a reasonable range of the parameter ζ. Unfortunately such calculations have not been published. Reference 16 provides curves for one typical example only. That example has the explosion strength $\zeta = 0.076$, i.e., it is a limiting case where the strong blast approximation of the shock strength is not valid. (The calculation is done for the density ratio $\rho_1/\rho_o = 1.16$, instead of 1.0, but this is probably not significant.) One can hypothesize that the approximation by strong blast formulas will be better for more powerful explosions than in this example.

A comparison between numerical flow field calculations and strong blast formulas is done in Reference 16 in terms of relative pressure profiles p/p_s, relative particle velocity profiles u/u_s and relative sound speed profiles c/c_s. The calculations and comparisons are presented for times corresponding to strong blast shock positions between $r/R = 3.58$ and $r/R = 7.60$. From Figure 7 one can see that for these positions the computed shock strength differs significantly from the strong blast shock strength. (See curve $\zeta = 0.08$ in Figure 7). Nevertheless, the corresponding relative pressure profiles differ by no more than 20%, the strong blast formulas providing higher values. (The comparison is made within a strip behind the leading shock covering 15% of the blast bubble radius). The differences become smaller for later times. At $r/R = 7.60$ the maximum difference is only about 1%.

The relative particle velocity profiles behave differently. Again, the strong blast formulas provide higher values. The difference is up to 13% and the approximation does not improve as time increases.

The relative sound speed profiles are up to 10% below the strong blast solution, and the difference does not decrease as the time increases.

The gas density is proportional tc c^{-2} and, therefore, the calculated density would be at large distances up to 20% higher than predicted by strong blast formulas. Consequently the calculated dynamic pressure ($\sim u^2 c^{-2}$) can be expected to be about 6% lower than given by the strong blast formulas, because the deviations in sound speed and particle velocity compensate each other.

In summary, in the example, presented in Reference 16, the relative incident pressure profiles and the relative dynamic pressure profiles can be reasonably approximated by corresponding strong blast profiles, particularly at large distances. This finding is encouraging because the example represents a case in which the initial shock pressure differs significantly from the corresponding strong blast pressure. One can expect better approximations for stronger explosions (smaller parameter ζ), but this hypothesis should be tested by sample calculations. Such calculations can be carried out, e.g., by using the computer program described in Reference 17.

REFERENCES

1. H.A. Bethe, K. Fuchs, J. von Neumann, R. Peierls, and W.G. Penny, U.S. Atomic Energy Commission Report AECO-2860, 1944.

2. J.L. Taylor, "An Exact Solution of the Spherical Blast Wave Problem", Phil. Mag., Vol 46, 1955.

3. R. Latter, "Similarity Solution for a Spherical Shock Wave", Journal of Applied Physics, Vol 26, No. 8 (1955).

4. L.I. Sedov, "Similarity and Dimensional Methods in Mechanics", Academic Press, New York (1959).

5. C.A. Rouse, "Theoretical Analysis of the Hydrodynamic Flow in Exploding Wire Phenomena, in "Exploding Wires", W.G. Chace and K.M. Howard, edts., Plenum Press, New York, pp. 227-263 (1959).

6. O. Laporte and T.S. Chang, "Exact Expressions for Curved Characteristics Behind Strong Blast Waves", U.S. Army Ballistic Research Laboratory Contractor Report, BRL-CR-30 (January 1971). (AD#722777)

7. O. Laporte and T.S. Chang, "Curved Characteristics Behind Blast Waves", Physics of Fluids, Vol 15, pp. 502-504 (1972).

8. P.C. Chou and R.R. Karpp, "Solution of Blast Waves by the Method of Characteristics", Drexel Institute of Technology Report, DIT Report No. 125-7, (1965).

9. J.A. Schmitt, "A Finite Element Method and Corresponding Pilot Computer Code for Hyperbolic Systems of Equations in Two Spatial Dimensions and Time Applied to Unsteady Gas Flows", U.S. Army Ballistic Research Laboratory Report, BRL-R-2017 (September 1977).(AD#A045703)

10. N. Gerber and J.M. Bartos, "Tables of Cylindrical Blast Functions for $\gamma = 5/3$ and $\gamma = 7/5$", U.S. Army Ballistic Research Laboratory Report, BRL-R-1376 (October 1961). (AD#663821)

11. J.W. Goresh and R.G. Dunn, "Tables of Blast Wave Parameters, I Spherical Explosions", Aerospace Research Laboratory Report, ARL-69-0011 (January 1969).

12. A. Sakurai, "Blast Wave Theory", Mathematics Research Center Technical Summary Report 497 (September 1964).

13. American National Standards Institute, American Society for Testing Materials, "Standard for Metric Practice", ASTM E380-76, ASTM, Philadelphia, PA, 1976.

14. Samuel Glasstone and Philip F. Dolan, eds, "The Effects of Nuclear Weapons", 3rd edition, U.S. Department of Defense and U.S. Department of Energy, 1977.

15. M. Lutzky and D. Lehto, "Transformations for Scaling of Close-In Pressures from Nuclear Explosions", Naval Ordnance Laboratory Report NOLTR-66-12, Silver Spring, MD, March 1966.

16. Shih Lien Huang and Pei Chi Chou, "Calculations of Expanding Shock Waves and Late-Stage Equivalence", Drexel Institute of Technology Report 125-12, Philadelphia, PA, April 1968.

17. Donald L. Tuckmantel, James Mao and Pei Chi Chou, "MCDU-8 A Computer Code for One-Dimensional Blast Wave Problems", Ballistic Reseach Laboratories Contract Report, BRL-CR-243, July 1975. (AD#A013864)

LIST OF SYMBOLS

a	constant defined by Eq. (2.16)
c	sound speed (m/s)
$C_1,\ldots,C_5$	constants defined by Eq. (2.17)
E	energy contained in a bursting sphere (J)
E_0	energy released by the explosion (J.m^{n-3})
$g(v)$	dimensionless function defined by Eq. (2.13)
$k(v)$	dimensionless function defined by Eq. (2.14)
$K(n,\gamma)$	proportionality factor, see Eq. (2.1) and (2.18)
M_+, M_-	factors in Mach line formulas (2.30)
n	dimension of the event ($n=1,2,3$ for planar, cylindrical, spherical events, respectively)
p	pressure (Pa)
p_0	ambient pressure (Pa)
p_1	initial pressure in a bursting sphere (Pa)
p_s	shock pressure (Pa)
p_{so}	initial shock pressure in a bursting sphere event (Pa)
p_{smax}	maximum shock pressure observed (Pa)
r	distance from the center of a bursting sphere
R	fireball radius (m)
t,T	time after the explosion (s)
$t_1(v),t_2(v)$	dimensionless functions defined by Eqs. (2.13) and (2.32)
t_s	shock arrival time (s)
u	particle velocity (m/s)
u_s	particle velocity immediately behind the shock (m/s)
U	shock velocity (m/s)

LIST OF SYMBOLS (Continued)

v	running parameter (see Eq. (2.15)
v_1, v_2	particular values of v, defined by Eq. (2.15)
$w(v)$	dimensionless function defined by Eq. (2.27)
x, X	distance from the center of the explosion (m)
x_s, X_s	shock distance from the center of the explosion (m)
$y(v)$	dimensionless function defined by Eq. (2.12)
γ	ratio of specific heats
ζ	explosion strength parameter defined by Eq. (5.2)
ρ	gas density (kg/m^3)
ρ_o	ambient gas density (kg/m^3)
ρ_1	initial gas density in a bursting sphere (kg/m^3)
ρ_s	gas density immediately behind the shock (kg/m^3)

APPENDIX

LISTING OF COMPUTER PROGRAMS

```
      SUBROUTINE SBLPREP(N,AIRP,AIRT,AIRG,AIRM,CHAREN,NBAD)
C     THIS IS PREPARATION ROUTINE FOR STRONG BLAST COMPUTATIONS
      COMMON/SBLCOM/NC,GAMMA,RO,CHEN,V(2),A,B(2),C(5),P(3),AK,NGOOD
      EXTERNAL SBLINTE
      DATA(NFIRST=0)
      IF(NFIRST.EQ.0) NGOOD=0
      NFIRST=1
      NBAD=0
      IDEF=0
      PRES=AIRP
      IF(PRES.GT.0.0)GOTO 15
      PRES=101325.0
      IDEF=IDEF+1
   15 TEMP=AIRT
      IF(TEMP.GT.0.0)GOTO 25
      TEMP=293.0
      IDEF=IDEF+1
   25 GAM=AIRG
      IF(GAM.GT.0.0)GOTO 35
      GAM=1.4
      IDEF=IDEF+1
   35 IF(GAM.NE.2.0)GOTO 38
      GAM=1.99
      IDEF=IDEF+1
   38 AMOL=AIRM
      IF(AMOL.GT.0.0)GOTO 65
      AMOL=0.02896
      IDEF=IDEF+1
   65 IF(CHAREN.GT.0.)GOTO 85
      NBAD=1
      PRINT 75,NBAD
      RETURN
   75 FORMAT(1H0,10X,30HRETURN FROM SBLPREP WITH NBAD=,I2,
     A39H BECAUSE CHARGE ENERGY IS NOT SPECIFIED)
   85 IF(N.GE.1.AND.N.LE.3)GOTO 101
      NBAD=2
      PRINT 95,NBAD,N
      RETURN
   95 FORMAT(1H ,10X,30HRETURN FROM SBLPREP WITH NBAD=,I2,
     A11H BECAUSE N=,I3,24H IS OUTSIDE RANGE 1 TO 3)
  101 IF(GAM.GT.1.0.AND.GAM.LT.7.0)GOTO 105
      NBAD=3
      PRINT 103,NBAD,GAM
      RETURN
  103 FORMAT(1H0,10X,30HRETURN FROM SBLPREP WITH NBAD=,I2,
     A 15H BECAUSE GAMMA=,1PE12.5,32H IS OUTSIDE THE RANGE 1.0 TO 7.0)
  105 NC=N
      AN=N
      GAMMA=GAM
      RO=PRES*AMOL/(8.3143*TEMP)
      CHEN=CHAREN
      V(1)=2.0/((2.0+AN)*GAMMA)
      V(2)=4.0/((2.0+AN)*(1.0+GAMMA))
      A=1.0+AN*(GAMMA-1.0)*0.5
      C(1)=2.0/(2.0+AN)
      C(2)=(GAMMA-1.0)/(2.0*GAMMA+AN-2.0)
      C(3)=2.0*AN*(GAMMA-2.0)/((2.0+AN)*(AN*GAMMA-AN+2.0))-GAMMA*
     A (2.0+AN)*(GAMMA-1.0)/((2.0*GAMMA+AN-2.0)*(AN*GAMMA-AN+2.0))      SBLPRE5
      C(4)=GAMMA/(GAMMA-2.0)                                            SBLPRE5
C  GAMMA=2.0 IS REPLACED BY 1.99 AT STATEMENT 35                        SBLPRE6
      C(5)=AN*(2.0*GAMMA+AN-2.0)/((2.0+AN)*(AN*GAMMA-AN+2.0))           SBLPRE6
     A+(2.0+AN)*GAMMA*(GAMMA-1.0)/( 2.0*(2.0-GAMMA)*(AN*GAMMA-AN+2.0))  SBLPRE62
      B(1)=(-1.0-GAMMA+2.0*GAMMA*A*V(1))*A*V(1)/(GAMMA-1.0)             SBLPRE63
      B(2)=(2.0-(GAMMA+1.0)*A*V(1))/(GAMMA-1.0)                         SBLPRE64
      DA=(2.+AN)*(AN*GAMMA**2-2.0*AN*GAMMA+AN+2.0*GAMMA-2.0)            SBLPRE65
      DB=2.0*(2.0*AN*GAMMA+2.0*GAMMA-AN+2.0)*(2.0-AN-2.0*GAMMA)         SBLPRE66
     A+2.0*AN*GAMMA*(2.0-GAMMA)*(AN*GAMMA+4.0)+2.0*(GAMMA+1.0)*DA       SBLPRE67
      P(2)=AN*GAMMA*(2.0+AN)*(2.0-GAMMA)*(GAMMA-1.0)/DB                 SBLPRE68
      P(1)=P(2)*(2.0*GAMMA+AN-2.0)/(AN*(GAMMA-2.0))                     SBLPRE69
      P(3)=P(2)*DA/(AN*(2.0-GAMMA)*(AN*GAMMA-AN+2.0))-1.0               SBLPRE70
C  NEXT START QUADRATURE TO COMPUTE AK                                  SBLPRE71
      CALL SBLROMB(SBLINTE,0.0,1.0,SF,NBAD)                             SBLPRE72
      IF(NBAD.EQ.0)GOTO 135                                             SBLPRE73
      PRINT 125,NBAD                                                    SBLPRE74
      RETURN                                                            SBLPRE75
  125 FORMAT(1H0,10X,30HRETURN FROM SBLPREP WITH NBAD=,I4,              SBLPRE76
     A28H BECAUSE OF ERROR IN SBLROMB)                                  SBLPRE77
  135 IF(N.EQ.1)FACT=1.0                                                SBLPRE78
      IF(N.EQ.2)FACT=3.14159265359                                      SBLPRE79
      IF(N.EQ.3)FACT=6.28318530718                                      SBLPRE80
      DAK=FACT*(2./(2.+AN))**2*(4./(GAMMA**2-1))*SF/(AN*C(2))           SBLPRE81
      AK=1.0/DAK**(1.0/(2.0+AN))                                        SBLPRE82
      NGOOD=1                                                           SBLPRE83
      PRINT145,NC,GAM,RO,CHEN,V(1),V(2),A,B(1),B(2),(C(J),J=1,5),P(1),  SBLPRE84
     AP(2),P(3),AK,NGOOD                                                SBLPRE85
  145 FORMAT(1H1,10X,26HCONTENTS OF COMMON/SBLCOM/,                     SBLPRE86
     X30H FOR STRONG BLAST CALCULATIONS,//,1H ,10X,3HNC=,               SBLPRE87
     AI2,5X,4HGAM=,0PF8.5,5X,4HRHO=,1PE12.5,5X,5HCHEN=,1PE12.5,//,      SBLPRE88
     Z 1H ,10X,5HV(J)=,2(2X,1PE12.5),//,                                SBLPRE89
     B 1H ,10X,2HA=,1PE12.5,/,1H0,10X,5HB(J)=,2(2X,1PE12.5),//,         SBLPRE90
     C 1H ,10X,5HC(J)=,5(2X,1PE12.5)                                    SBLPRE91
     D//,1H ,10X,5HP(J)=,3(2X,1PE12.5),//,                              SBLPRE92
     E 1H ,10X,3HAK=,1PE12.5,//,1H ,10X,6HNGOOD=,I2,//)                 SBLPRE93
      IF(IDEF.EQ.0)GOTO 165                                             SBLPRE94
      PRINT155,PRES,TEMP,GAM,AMOL                                       SBLPRE95
  155 FORMAT(1H0,10X,41HSOME OF THE FOLLOWING ARE DEFAULT VALUES ,/,1H ,SBLPRE96
     A10X,44HASSIGNED BY SBLPREP AND CORRESPONDING TO AIR,//,1H ,10X,   SBLPRE97
     C13HPRESSURE    =,1PE12.5,3H PA,/,                                 SBLPRE98
     D1H ,10X,13HTEMPERATURE =,1PE12.5,2H K,/,                          SBLPRE99
     D1H ,10X,13HGAMMA       =,1PE12.5,/,                               SBLPR100
     E1H ,10X,13HMOLAR MASS  =,1PE12.5,8H KG/MOLE)                      SBLPR101
  165 RETURN                                                            SBLPR102
      END                                                               SBLPR103
      SUBROUTINE SBLROMB(F,A,B,FINT,NBAD)
C     ROMBERG INTEGRATION SUBROUTINE
      DIMENSION T(15,20)
      NBAD=0
      CALL F(A,FA,NBAD)
      IF(NBAD.NE.0)RETURN
      CALL F(B,FB,NBAD)
      IF(NBAD.NE.0)RETURN
      T(1,1)=(FA+FB)*0.5
      KM=1
      KMA=1
   15 DEN=FLOAT(KMA)*2.
      FM=0
      DO 25 KA=1,KMA
      AC=FLOAT(1+2*(KMA-KA))/DEN
      BC=FLOAT(2*KA-1)/DEN
      ARG=AC*A+BC*B
      IF(B-A) 17,18,19
   17 ARG=AMAX1(ARG,B)
      ARG=AMIN1(ARG,A)
      GOTO 20
   18 FINT=0.
      RETURN
   19 ARG=AMAX1(ARG,A)
      ARG=AMIN1(ARG,B)
   20 CALL F(ARG,FN,NBAD)
      IF(NBAD.NE.0)RETURN
      FM=FM+FN
   25 CONTINUE
      FM=FM/FLOAT(KMA)
      T(1,KM+1)=(T(1,KM)+FM)*0.5
C     THIS IS TRAPEZ. NOW COMPUTE ROMBERG
      KM=KM+1
      KC=1
      DDEN=1.
   35 KC=KC+1
      DDEN=DDEN*4.
      T(KC,KM)=T(KC-1,KM)+(T(KC-1,KM)-T(KC-1,KM-1))/(DDEN-1.)
      IF(KC.LT.KM.AND.KC.LT.15)GOTO 35
      IF(KC.GE.4)GOTO 45
      KMA=KMA*2
      GOTO 15
   45 KAF=KC-3
      ITEST=0
      DO 55 KA=KAF,KC
C     NOW TEST CONVERGENCE
      TEST=ABS(T(KA,KM)-T(KA-1,KM-1))
      IF(TEST.GT.ABS(T(KC,KM))*1.E-5 ) ITEST=1
      IF(TEST.LE.1.E-100)GOTO 65
   55 CONTINUE
      IF(ITEST.EQ.0) GOTO 65
      IF(KM.GE.20)GOTO 65
      KMA=KMA*2
      GOTO 15
   65 FINT=T(KC,KM)*(B-A)
      RETURN
      END
      SUBROUTINE SBLINTE(W,F,NBAD)
C  THIS IS THE INTEGRAND ROUTINE FOR THE COMPUTATION OF AK
C  THE INTEGRAND DOES NOT CONTAIN THE PI-FACTOR ETC.
      COMMON/SBLCOM/N,GAM,RO,CHEN,V(2),A,B(2),C(5),P(3),AK,NGOOD
      NBAD=0
      EX1=2.+C(1)*FLOAT(2+N)
      EX2=C(2)*FLOAT(N)
      EX3=C(5)*2.+C(3)*FLOAT(N)
      EX4=C(4)
      IF(W-1.0) 12,27,15
   12 IF(W) 15,25,35
   15 NBAD=1
      RETURN
   25 U=V(1)
      GOTO 37
   27 U=V(2)
      GOTO 37
   35 U=V(1)+(V(2)-V(1))*W**(1./EX2)
   37 D1=U/V(2)
      D2=(U-V(1))/(V(2)-V(1))
      D3=(1.0-A*U)/(1.0-A*V(2))
      D4=(V(1)*GAM-U)/(V(1)*GAM-V(2))
      D5=C(2)-(U-V(1))*(C(1)/U+C(3)*A/(1.0-A*U))
      F=(D1**EX1*D2/D4+1.0)*D3**EX3*D4**EX4*D5
      RETURN
      END
      SUBROUTINE SBLSHCK(RMIN,RMAX,NR,R,T,US,P,UP,RO,DP,NBAD)
C   THIS COMPUTES STRONG SHOCK BETWEEN RMIN AND RMAX
C   NR = NUMBER OF NODES TO BE COMPUTED
C   R = DISTANCE
C   T = SHOCK ARRIVAL TIME
C   US = SHOCK VELOCITY
C   P = INCIDENT SHOCK PRESSURE
C   UP = PARTICLE VELOCITY
C   RO = DENSITY
C   DP = DYNAMIC PRESSURE = 0.5*RO*UP**2
C   NBAD = ERROR INDICATED BY NBAD.NE.0
      DIMENSION R(1),T(1),US(1),P(1),UP(1),RO(1),DP(1)
      COMMON/SBLCOM/NC,GAM,ROC,CHEN,V(2),A,B(2),C(5),PC(3),AK,NGOOD
      IF(NGOOD.NE.0) GOTO 10
      NBAD=11
      PRINT 11,NBAD
      RETURN
   11 FORMAT(1H0,10X,30HRETURN FROM SBLSHCK WITH NBAD=,I3,
     A26H AND WITHOUT COMPUTATIONS.,/,1H ,10X,
     B34HSUBROUTINE SBLPREP MUST BE CALLED ,
     C39H BEFORE OTHER SBL-ROUTINES CAN BE USED.,/)
   10 NBAD=0
      IF(NR.GE.1)GOTO 25
      NBAD=1
      PRINT 15,NBAD,NR
      RETURN
   15 FORMAT(1H0,10X,30HRETURN FROM SBLSHCK WITH NBAD=,I2,
     A12H BECAUSE NR=,I4)
   25 R(1)=AMAX1(RMIN,0.)
      ANT=2+NC
      EX=ANT/2.0
      TFACT=SQRT(ROC/CHEN)/AK**EX
      ROFACT=((GAM+1.0)/(GAM-1.0))*ROC
      DO 55 KA=1,NR
      IF(R(KA).GT.0.)GOTO 35
      T(KA)=0.
      US(KA)=0
      P(KA)=0
      DP(KA)=0
C   SINGULAR VALUES AT R=0 REPLACED BY ZERO
      GOTO 41
   35 T(KA)=TFACT*R(KA)**EX
      US(KA)=R(KA)/(T(KA)*EX)
      P(KA)=2.0*ROC*US(KA)**2/(1.0+GAM)
      UP(KA)=2.0*US(KA)/(1.0+GAM)
      RO(KA)=ROFACT
      DP(KA)=ROFACT*UP(KA)**2*0.5
   41 IF(KA.EQ.NR)GOTO 55
      R(KA+1)=R(KA)+(RMAX-R(1))/FLOAT(NR-1)
      IF(R(KA+1).GT.R(KA))GOTO 55
      NBAD=2
      PRINT 45,NBAD,RMAX,RMIN
      RETURN
   45 FORMAT(1H0,10X,30HRETURN FROM SBLSHCK WITH NBAD=,I2,
     A14H BECAUSE RMAX=,1PE12.5,20H  IS NOT LARGER THAN,
     B6H RMIN=,1PE12.5)
   55 CONTINUE
```

```
      RETURN                                                            SBLSHC58
      END                                                               SBLSHC59
      SUBROUTINE SBLPROF(T,RMIN,RMAX,NR,R,P,UP,RO,DP,NBAD)
C   THIS COMPUTES THE FLOW PROFILE AT TIME T BETWEEN RMIN AND RMAX
C   T = TIME AFTER THE EXPLOSION
C   RMIN,RMAX = PROFILE LIMITS
C   NR = NUMBER OF MODES TO BE COMPUTED
C   R = DISTANCE FROM THE EXPLOSION
C   P = PRESSURE
C   UP = PARTICLE VELOCITY
C   RO = DENSITY
C   DP = DYNAMIC PRESSURE = 0.5*RO*UP**2
C   NBAD = ERROR RETURN INDICATED BY NBAD.NE.0
      DIMENSION R(1),P(1),UP(1),RO(1),DP(1)
      COMMON/SBLCOM/NC,GAM,ROC,CHEN,V(2),A,B(2),C(5),PC(3),AK,NGOOD
      Y(D)=(V(2)/D)**C(1)*((D-V(1))/(V(2)-V(1)))**C(2)
     A   *((1.+A*D)/(1.+A*V(2)))**C(3)
      G(D)=(D/V(2))**(C(1)*FLOAT(NC))*((V(1)*GAM-D)/(V(1)*GAM-V(2)))**C(
     A   4)*((1.+A*D)/(1.+A*V(2)))**(2.*C(5))
      H(D)=((D-V(1))/(V(2)-V(1)))**(1.+2.*C(2))
     A   *((V(1)*GAM-D)/(V(1)*GAM-V(2)))**(C(4)-1.)
     B   *((1.+A*D)/(1.+A*V(2)))**(2.*C(5)-2.*C(3))
      U(D)=V(1)+(V(2)-V(1))*D**(1./C(2))
      IF(NGOOD.NE.0) GOTO 10
      NBAD=11
      PRINT 11,NBAD
      RETURN
   11 FORMAT(1H0,10X,30HRETURN FROM SBLPROF WITH NBAD=,I3,
     A26H AND WITHOUT COMPUTATIONS.,/,1H ,10X,
     B34HSUBROUTINE SBLPREP MUST BE CALLED ,
     C39H BEFORE OTHER SBL-ROUTINES CAN BE USED.,/)
   10 NBAD=0
      IF(T.GT.0.0)GOTO 25
      NBAD=1
      PRINT 15,NBAD,T
      RETURN
   15 FORMAT(1H0,10X,30HRETURN FROM SBLPROF WITH NBAD=,I2,
     A11H BECAUSE T=,1PE12.5)
   25 IF(NR.GE.1)GOTO 45
      NBAD=2
      PRINT 35,NBAD,NR
      RETURN
   35 FORMAT(1H0,10X,30HRETURN FROM SBLPROF WITH NBAD=,I2,
     A12H BECAUSE NR=,I4)
   45 ANT=NC+2
      RS=AK*(CHEN/ROC)**(1./ANT)*T**(2./ANT)
      USHCK=2.0*RS/(T*ANT)
      US=2.0*USHCK/(1.+GAM)
      PS=2.0*ROC*USHCK**2/(1.+GAM)
      ROS=(GAM+1.0)*ROC/(GAM-1.)
      RMI=AMAX1(RMIN,0.0)
      RMI=AMIN1(RMI,RS)
      RMA=AMIN1(RMAX,RS)
C   NEXT FIND PARAMETER V CORRESPONDING TO RMI
      X1=0.0
      F1=0.0
      X2=1.0
      F2=1.0
      ICNT=0
      IF(RMI.GT.0.0)GOTO 55                                             SBLPR058
      VMI=0.0                                                           SBLPR059
      GOTO 85                                                           SBLPR060
   55 IF(RMI.LT.RS)GOTO 65                                              SBLPR061
      VMI=1.0                                                           SBLPR062
      GOTO 35                                                           SBLPR063
   65 X3=X2-(X2-X1)*(F2-RMI/RS)/(F2-F1)                                 SBLPR064
      XW=W(X3)                                                          SBLPR065
      F3=Y(XW)                                                          SBLPR066
      IF(ABS(F3-RMI/RS).LT.1.0E-7*RMI/RS) GOTO 75                       SBLPR067
      ICNT=ICNT+1                                                       SBLPR068
      IF(ICNT.GT.20)GOTO 75                                             SBLPR069
      IF(F3-RMI/RS) 56,75,58                                            SBLPR070
   56 X1=X3                                                             SBLPR071
      F1=F3                                                             SBLPR072
      GOTO 65                                                           SBLPR073
   58 X2=X3                                                             SBLPR074
      F2=F3                                                             SBLPR075
      GOTO 65                                                           SBLPR076
   75 VMI=X3                                                            SBLPR077
   85 R(1)=RS*Y(W(VMI))                                                 SBLPR078
      P(1)=PS*G(W(VMI))                                                 SBLPR079
      UP(1)=US*(R(1)/RS)*(W(VMI) /V(2))                                 SBLPR080
      RO(1)=ROS*H(W(VMI))                                               SBLPR081
      DP(1)=RO(1)*UP(1)**2*0.5                                          SBLPR082
      IF(NR.EQ.1)RETURN                                                 SBLPR083
      IF(RMA.GT.RMI)GOTO 105                                            SBLPR084
      NBAD=3                                                            SBLPR085
      PRINT 95,NBAD,RMAX                                                SBLPR086
      RETURN                                                            SBLPR087
   95 FORMAT(1H0,10X,30HRETURN FROM SBLPROF WITH NBAD=,I2,              SBLPR088
     A14H BECAUSE RMAX=,1PE12.5,18H  IS OUTSIDE RANGE)                  SBLPR089
  105 IF(RMA.LT.RS)GOTO 115                                             SBLPR090
      VMA=1.00                                                          SBLPR091
      GOTO 145                                                          SBLPR092
C  NOW COMPUTE PARAMETER VMA CORRESPONDING TO RMA                       SBLPR093
  115 ICNT=0                                                            SBLPR094
      X1=VMI                                                            SBLPR095
      F1=Y(W(X1))                                                       SBLPR096
      X2=1.0                                                            SBLPR097
      F2=1.0                                                            SBLPR098
  125 X3=X2-(X2-X1)*(F2-RMA/RS)/(F2-F1)                                 SBLPR099
      XW=W(X3)                                                          SBLPR100
      F3=Y(XW)                                                          SBLPR101
      IF(ABS(F3-RMA/RS).LT.1.0E-7 *RMA/RS) GOTO 135                     SBLPR102
      ICNT=ICNT+1                                                       SBLPR103
      IF(ICNT.GT.20)GOTO 135                                            SBLPR104
      IF(F3-RMA/RS)126,135,128                                          SBLPR105
  126 X1=X3                                                             SBLPR106
      F1=F3                                                             SBLPR107
      GOTO 125                                                          SBLPR108
  128 X2=X3                                                             SBLPR109
      F2=F3                                                             SBLPR110
      GOTO 125                                                          SBLPR111
  135 VMA=X3                                                            SBLPR112
  145 DO 155 KA=2,NR                                                    SBLPR113
      VK=VMI+(VMA-VMI)*FLOAT(KA-1)/FLOAT(NR-1)                          SBLPR114
      VK=W(VK)                                                          SBLPR115
      R(KA)=RS*Y(VK)                                                    SBLPR116
      P(KA)=PS*G(VK)                                                    SBLPR117
      UP(KA)=US*(R(KA)/RS)*(VK/V(2))                                    SBLPR118
      RO(KA)=ROS*H(VK)                                                  SBLPR119
      DP(KA)=RO(KA)*UP(KA)**2*0.5                                       SBLPR120
  155 CONTINUE                                                          SBLPR121
      RETURN                                                            SBLPR122
      END                                                               SBLPR123
      SUBROUTINE SBLHIST(R,TMIN,TMAX,NR,T,P,UP,RO,DP,NBAD)              ****    1
C     THIS COMPUTES THE FLOW HISTORY AT DISTANCE R AND TIME BETWEEN TMIN,TMSBLHIS 2
C     R = DISTANCE                                                      SBLHIS 3
C     TMIN, TMAX = HISTORY LIMITS                                       SBLHIS 4
C     NR = NUMBER OF NODES TO BE COMPUTED                               SBLHIS 5
C     T = TIME                                                          SBLHIS 6
C     P = PRESSURE                                                      SBLHIS 7
C     UP = PARTICLE VELOCITY                                            SBLHIS 8
C     RO = DENSITY                                                      SBLHIS 9
C     DP = DYNAMIC PRESSURE = 0.5*RO*UP**2                              SBLHIS10
C     NBAD = ERROR RETURN IS INDICATED BY NBAD.NE.0                     SBLHIS11
      DIMENSION T(1),P(1),UP(1),RO(1),DP(1)                             SBLHIS12
      COMMON/SBLCOM/NC,GAM,ROC,CHEN,V(2),A,B(2),C(5),PC(3),AK,NGOOD     SBLHIS13
      Y(D)=(V(2)/D)**C(1)*((D-V(1))/(V(2)-V(1)))**C(2)                  SBLHIS14
     A   *((1.+D*A)/(1.+V(2)*A))**C(3)                                  SBLHIS15
      G(D)=(D/V(2))**(C(1)*FLOAT(NC))*((V(1)*GAM-D)/(V(1)*GAM-V(2)))**C(SBLHIS16
     A   4)*((1.+D*A)/(1.+V(2)*A))**(2.*C(5))                           SBLHIS17
      H(D)=((D-V(1))/(V(2)-V(1)))**(1.-2.*C(2))                         SBLHIS18
     A   *((V(1)*GAM-D)/(V(1)*GAM-V(2)))**(C(4)-1.)                     SBLHIS19
     B   *((1.+D*A)/(1.+V(2)*A))**(2.*C(5)-2.*C(3))                     SBLHIS20
      W(D)=V(1)+(V(2)-V(1))*D**(1./C(2))                                SBLHIS21
      IF(NGOOD.NE.0) GOTO 10                                            SBLHIS22
      NBAD=11                                                           SBLHIS23
      PRINT 11,NBAD                                                     SBLHIS24
      RETURN                                                            SBLHIS25
   11 FORMAT(1H0,10X,30HRETURN FROM SBLHIST WITH NBAD=,I3,              SBLHIS26
     A26H AND WITHOUT COMPUTATIONS.,/,1H ,10X,                          SBLHIS27
     B34HSUBROUTINE SBLPREP MUST BE CALLED ,                            SBLHIS28
     C39H BEFORE OTHER SBL-ROUTINES CAN BE USED.,/)                     SBLHIS29
   10 NBAD=0                                                            SBLHIS30
      IF(R.GT.0.)GOTO 25                                                SBLHIS31
      NBAD=1                                                            SBLHIS32
      PRINT 15,NBAD,R                                                   SBLHIS33
      RETURN                                                            SBLHIS34
   15 FORMAT(1H0,10X,30HRETURN FROM SBLHIST WITH NBAD=,I2,              SBLHIS35
     A11H BECAUSE R=,1PE12.5)                                           SBLHIS36
   25 IF(NR.GE.1)GOTO 27                                                SBLHIS37
      NBAD=2                                                            SBLHIS38
      PRINT 26,NBAD,NR                                                  SBLHIS39
      RETURN                                                            SBLHIS40
   26 FORMAT(1H0,10X,30HRETURN FROM SBLHIST WITH NBAD=,I2,              SBLHIS41
     A12H BECAUSE NR=,I4)                                               SBLHIS42
   27 ANT=0.5*FLOAT(2+NC)                                               SBLHIS43
      ROS=(GAM+1.)*ROC/(GAM-1.)                                         SBLHIS44
      TS=SQRT(ROC/CHEN)*(R/AK)**ANT                                     SBLHIS45
C     THIS IS SHOCK ARRIVAL TIME AT R                                   SBLHIS46
      TMI=AMAX1(TMIN,TS)                                                SBLHIS47
      IF(TMI.GT.TS)GOTO 35                                              SBLHIS48
      VMI=1.0                                                           SBLHIS49
      GOTO 65                                                           SBLHIS50
C     NEXT COMPUTE PARAMETER VMI CORRESPONDING TO TMI                   SBLHIS51
   35 ICNT=0                                                            SBLHIS52
      DF=(TS/TMI)**(1./ANT)                                             SBLHIS53
      X1=0.0                                                            SBLHIS54
      F1=0                                                              SBLHIS55
      X2  =1.0                                                          SBLHIS56
      F2=1.                                                             SBLHIS57
   45 X3=X2-(X2-X1)*(F2-DF)/(F2-F1)
      XW=W(X3)
      F3=Y(XW)
      IF(ABS(F3-DF).LT.1.0E-7 *DF)GOTO 55
      ICNT=ICNT+1
      IF(ICNT.GT.20)GOTO 55
      IF(F3-DF)46,55,48
   46 X1=X3
      F1=F3
      GOTO 45
   48 X2=X3
      F2=F3
      GOTO 45
   55 VMI=X3
   65 XW=W(VMI)
      YKA=Y(XW)
      T(1)=TS*YKA**(-ANT)
      USHCK=(R/YKA)/(T(1)*ANT)
      P(1)=2.*(ROC/(1.+GAM))*USHCK**2*G(XW)
      UP(1)=R*XW/T(1)
      RO(1)=ROS*H(XW)
      DP(1)=RO(1)*UP(1)**2*0.5
      IF(NR.EQ.1)RETURN
      IF(TMAX.GT.TMI)GOTO 85
      NBAD=3
      PRINT 75,NBAD,TMAX
      RETURN
   75 FORMAT(1H0,10X,30HRETURN FROM SBLHIST WITH NBAD=,I4,
     A14H BECAUSE TMAX=,1PE12.5,17H IS OUTSIDE RANGE)
   85 ICNT=0
      DF=(TS/TMAX)**(1./ANT)
C    NOW FIND VMA CORRESPONDING TO TIME=TMAX
      X1=VMI
      F1=Y(W(VMI))
      X2=0.0
      F2=0.0
   95 X3=X2-(X2-X1)*(F2-DF)/(F2-F1)
      XW=W(X3)
      F3=Y(XW)
      IF(ABS(F3-DF).LT.DF*1.0E-7 )GOTO 105
      ICNT=ICNT+1
      IF(ICNT.GT.20)GOTO 105
      IF(F3-DF)96,105,98
   96 X2=X3
      F2=F3
      GOTO 95
   98 X1=X3
      F1=F3
      GOTO 95
  105 VMA=X3
      DO 115 KA=2,NR
      VK=VMI+(VMA-VMI)*FLOAT(KA-1)/FLOAT(NR-1)
      XW=W(VK)
      YKA=Y(XW)
      T(KA)=TS*YKA**(-ANT)
      RS=R/YKA
      USHCK=RS/(T(KA)*ANT)
      P(KA)=2.*(ROC/(1.+GAM))*USHCK**2*G(XW)
      UP(KA)=R*XW/T(KA)
      RO(KA)=ROS*H(XW)
      DP(KA)=RO(KA)*UP(KA)**2*0.5
  115 CONTINUE
      RETURN
      END
      SUBROUTINE SBLPATH(RZ,RMAX,TMAX,NR,R,T,P,UP,RO,DP,NBAD)                 ****    1
C   THIS COMPUTES A PARTICLE PATH STARTING ON THE SHOCK AT R=RZ             SBLPAT 2
C   RZ = DISTANCE OF FIRST NODE FROM EXPLOSION, LOCATED ON THE SHOCK        SBLPAT 3
C   RMAX,TMAX = PATH ENDS WHEN EITHER OF THESE IS REACHED                   SBLPAT 4
C   NR = NUMBER OF NODES TO BE COMPUTED                                     SBLPAT 5
C   R = DISTANCE FROM THE EXPLOSION                                         SBLPAT 6
C   T = TIME AFTER EXPLOSION                                                SBLPAT 7
C   P = PRESSURE                                                            SBLPAT 8
C   UP = PARTICLE VELOCITY                                                  SBLPAT 9
C   RO = DENSITY                                                            SBLPAT10
C   DP = DYNAMIC PRESSURE = 0.5*RO*UP**2                                    SBLPAT11
C   NBAD = ERROR RETURN WILL BE INDICATED BY NBAD.NE.0                      SBLPAT12
      DIMENSION R(1),T(1),P(1),UP(1),RO(1),DP(1)                            SBLPAT13
      COMMON/SBLCON/NC,GAM,ROC,CHEN,V(2),A,B(2),C(5),PC(3),AK,NGOOD         SBLPAT14
      Y(D)=(V(2)/D)**C(1)*((D-V(1))/(V(2)-V(1)))**C(2)                      SBLPAT15
     A    *((1.+D*A)/(1.+V(2)*A))**C(3)                                     SBLPAT16
      G(D)=(D/V(2))**(C(1)*FLOAT(NC))*((V(1)*GAM-D)/(V(1)*GAM-V(2)))**C(SBLPAT17
     A    4)*((1.+D*A)/(1.+V(2)*A))**(2.*C(5))                              SBLPAT18
      H(D)=((D-V(1))/(V(2)-V(1)))**(1.-2.*C(2))                             SBLPAT19
     A    *((V(1)*GAM-D)/(V(1)*GAM-V(2)))**(C(4)-1.)                        SBLPAT20
     B    *((1.+D*A)/(1.+V(2)*A))**(2.*C(5)-2.*C(3))                        SBLPAT21
      W(D)=(D/V(2))*((V(1)*GAM-D)/(V(1)*GAM-V(2)))**PC(1)                   SBLPAT22
     A    *((D-V(1))/(V(2)-V(1)))**PC(2)*((1.+D*A)/(1.+V(2)*A))**PC(3)      SBLPAT23
      RSF(D)=AK*(CHEN/ROC)**(1./FLOAT(2+NC))*D**(2./FLOAT(2+NC))            SBLPAT24
      X(D)=V(1)+(V(2)-V(1))*D**(-1./PC(2))                                  SBLPAT25
      IF(NGOOD.NE.0) GOTO 10                                                SBLPAT26
      NBAD=11                                                               SBLPAT27
      PRINT 11,NBAD                                                         SBLPAT28
      RETURN                                                                SBLPAT29
   11 FORMAT(1H0,10X,30HRETURN FROM SBLPATH WITH NBAD=,I3,                  SBLPAT30
     A26H AND WITHOUT COMPUTATIONS.,/,1H ,10X,                              SBLPAT31
     B34HSUBROUTINE SBLPREP MUST BE CALLED ,                                SBLPAT32
     C39H BEFORE OTHER SBL-ROUTINES CAN BE USED.,/)                         SBLPAT33
   10 NBAD=0                                                                SBLPAT34
      IF(RZ.GT.0.)GOTO 25                                                   SBLPAT35
      NBAD=1                                                                SBLPAT36
      PRINT 15,NBAD,RZ                                                      SBLPAT37
      RETURN                                                                SBLPAT38
   15 FORMAT(1H0,10X,30HRETURN FROM SBLPATH WITH NBAD=,I2,                  SBLPAT39
     A12H BECAUSE RZ=,1PE12.5)                                              SBLPAT40
   25 IF(NR.GT.0)GOTO 45                                                    SBLPAT41
      NBAD=2                                                                SBLPAT42
      PRINT 35,NBAD,NR                                                      SBLPAT43
      RETURN                                                                SBLPAT44
   35 FORMAT(1H0,10X,30HRETURN FROM SBLPATH WITH NBAD=,I2,                  SBLPAT45
     A12H BECAUSE NR=,I4)                                                   SBLPAT46
   45 R(1)=RZ                                                               SBLPAT47
      ANT=2+NC                                                              SBLPAT48
      T(1)=AK**(-0.5*ANT)*SQRT(ROC/CHEN)*RZ**(0.5*ANT)                      SBLPAT49
      USHCK=(2.0/ANT)*RZ/T(1)                                               SBLPAT50
      P(1)=(2.0/(GAM+1.0))*ROC*USHCK**2                                     SBLPAT51
      UP(1)=(2.0/(GAM+1.))*USHCK                                            SBLPAT52
      RO(1)=((GAM+1.)/(GAM-1.))*ROC                                         SBLPAT53
      DP(1)=RO(1)*UP(1)**2*0.5                                              SBLPAT54
      IF(NR.EQ.1)RETURN                                                     SBLPAT55
      IF(TMAX.GT.T(1).AND.RMAX.GT.R(1))GOTO 65                              SBLPAT56
      NBAD=3                                                                SBLPAT57
      PRINT 55,NBAD,TMAX,T(1),RMAX,R(1)                                 SBLPAT58
      RETURN                                                            SBLPAT59
   55 FORMAT(1H0,10X,30HRETURN FROM SBLPATH WITH NBAD=,I2,              SBLPAT60
     A14H BECAUSE TMAX=,1PE12.5,13H .LE. TSHOCK=,1PE12.5,/,             SBLPAT61
     B1H ,10X,16HOR BECAUSE RMAX=,1PE12.5,9H .LE. RZ=,1PE12.5)          SBLPAT62
   65 ICNT=0                                                            SBLPAT63
C   NOW COMPUTE PARAMETER VMA CORRESPONDING TO TMAX                     SBLPAT64
      DT=T(1)/TMAX                                                      SBLPAT65
      X1=0.0                                                            SBLPAT66
      F1=0.0                                                            SBLPAT67
      X2=1.0                                                            SBLPAT68
      F2=1.0                                                            SBLPAT69
   75 X3=X2-(X2-X1)*(F2-DT)/(F2-F1)                                     SBLPAT70
      F3=1./W(X(X3))                                                    SBLPAT71
      IF(ABS(F3-DT).LT.1.0E-7*DT) GOTO 85                               SBLPAT72
      ICNT=ICNT+1                                                       SBLPAT73
      IF(ICNT.GT.20)GOTO 85                                             SBLPAT74
      IF(F3-DT) 76,85,78                                                SBLPAT75
   76 X1=X3                                                             SBLPAT76
      F1=F3                                                             SBLPAT77
      GOTO 75                                                           SBLPAT78
   78 X2=X3                                                             SBLPAT79
      F2=F3                                                             SBLPAT80
      GOTO 75                                                           SBLPAT81
   85 VTMAX=X3                                                          SBLPAT82
      RTMAX=RSF(TMAX)*Y(X(VTMAX))                                       SBLPAT83
      IF(RTMAX.LT.RMAX)GOTO 105                                         SBLPAT84
C     BRANCH IF END OF PATH DETERMINED BY TMAX                          SBLPAT85
C     ELSE COMPUTE PARAMETER CORRESPONDING TO RMAX                      SBLPAT86
      ICNT=0                                                            SBLPAT87
      X1=VTMAX                                                          SBLPAT88
      F1=RTMAX                                                          SBLPAT89
      X2=1.0                                                            SBLPAT90
      F2=RZ                                                             SBLPAT91
   95 X3=X2-(X2-X1)*(F2-RMAX)/(F2-F1)                                   SBLPAT92
      DT=T(1)*W(X(X3))                                                  SBLPAT93
      F3=RSF(DT)*Y(X(X3))                                               SBLPAT94
      IF(ABS(F3-RMAX).LT.1.0E-7*RMAX) GOTO 105                          SBLPAT95
      ICNT=ICNT+1                                                       SBLPAT96
      IF(ICNT.GT.20)GOTO 105                                            SBLPAT97
      IF(F3-RMAX) 96,105,98                                             SBLPAT98
   96 X2=X3                                                             SBLPAT99
      F2=F3                                                             SBLPA100
      GOTO 95                                                           SBLPA101
   98 X1=X3                                                             SBLPA102
      F1=F3                                                             SBLPA103
      GOTO 95                                                           SBLPA104
  105 YMAX=1./X3                                                        SBLPA105
      YMIN=1.0                                                          SBLPA106
      DO 135 KA=2,NR                                                    SBLPA107
      YK=YMIN+(YMAX-YMIN)*FLOAT(KA-1)/FLOAT(NR-1)                       SBLPA108
      VK=V(1)+(V(2)-V(1))*YK**(1./PC(2))                                SBLPA109
      T(KA)=T(1)*W(VK)                                                  SBLPA110
      RS=RSF(T(KA))                                                     SBLPA111
      R(KA)=RS*Y(VK)                                                    SBLPA112
      USHCK=(RS/T(KA))*2.0/FLOAT(2+NC)                                  SBLPA113
      P(KA)=(2.0/(1.+GAM))*ROC*USHCK**2*G(VK)                           SBLPA114
      UP(KA)=R(KA)*VK/T(KA)                                             SBLPA115
      RO(KA)=((GAM+1.0)/(GAM-1.))*ROC*H(VK)                             SBLPA116
      DP(KA)=RO(KA)*UP(KA)**2*0.5                                       SBLPA117
  135 CONTINUE                                                          SBLPA118
      RETURN                                                            SBLPA119
      END                                                               SBLPA120
      SUBROUTINE SBLMACH(RZ,TZ,RMIN,RMAX,TMIN,TMAX,NR,
     A R,T,P,UP,RO,DP,NBAD)
C  THIS COMPUTES MACH-LINES AND CORRESPONDING FLOW FIELD
C  RZ,TZ = INITIAL POINT OF THE MACH LINES. EITHER RZ OR TZ MUST
C          BE ZERO. IF RZ=0, THEN COMPUTATION WILL START
C          AT THE EXPLOSION AND TIME TZ.
C          IF TZ=0, THEN COMPUTATION WILL START ON THE
C          SHOCK AT DISTANCE RZ AND CORRESPONDING TIME
C  RMIN,---,TMAX = LIMITS FOR THE COMPUTATIONS
C  NR(2)    = NUMBER OF NODES TO BE COMPUTED.
C             NR(1) CORRESPONDS TO CHARACTERISTIC WITH DT/DR.GT.0
C  R(2,1)   = DISTANCE   FROM THE EXPLOSION
C  T(2,1)   = TIME FROM THE EXPLOSION
C  P(2,1)   = PRESSURE
C  UP(2,1)  = PARTICLE VELOCITY
C  RO(2,1)  = DENSITY
C  DP(2,1)  = DYNAMIC PRESSURE = 0.5*RO*UP**2
C  NBAD     = ERROR RETURN WILL BE INDICATED BY NBAD.NE.0
      DIMENSION NR(2),R(2,1),T(2,1),P(2,1),RO(2,1),DP(2,1),UP(2,1)
      COMMON/SBLCOM/NC,GAM,ROC,CHEN,V(2),A,B(2),C(5),PC(3),AK,NGOOD
      Y(D)=(V(2)/D)**C(1)*((D-V(1))/(V(2)-V(1)))**C(2)
     A   *((1.+D*A)/(1.+V(2)*A))**C(3)
      G(D)=(D/V(2))**(C(1)*FLOAT(NC))*((V(1)*GAM-D)/(V(1)*GAM-V(2)))**C(
     A   4)*((1.-D*A)/(1.+V(2)*A))**(2.*C(5))
      H(D)=((D-V(1))/(V(2)-V(1)))**(1.-2.*C(2))
     A   *((V(1)*GAM-D)/(V(1)*GAM-V(2)))**(2.*C(5)-2.*C(3))
      RSF(D)=AK*((CHEN/ROC)*D**2)**(1./FLOAT(2+NC))
      Z1(D)=A*D/(1.+A*D)
      ZA(D)=(B(1)+B(2)*D*A)/((1.0-D*A)*V(1)*A)
      ZB(D)=AMAX1(-1.0,AMIN1(1.0,ZA(D)))
      Z2(D)=EXP(ASIN(ZB(D))/SQRT(B(1)+B(2)))
      XF(D)=V(1)+(V(2)-V(1))*D**(1./C(2))
      IF(NGOOD.NE.0) GOTO 10
      NBAD=11
      PRINT 11,NBAD
      RETURN
   11 FORMAT(1H0,10X,30HRETURN FROM SBLMACH WITH NBAD=,I3,
     A26H AND WITHOUT COMPUTATIONS.,/,1H ,10X,
     B34HSUBROUTINE SBLPREP MUST BE CALLED ,
     C39H BEFORE OTHER SBL-ROUTINES CAN BE USED.,/)
   10 NBAD=0
      Z2V1=EXP(-1.5707963268/SQRT(B(1)+B(2)))
      IF(NR(1).GE.1.OR.NR(2).GE.1)GOTO 25
      NBAD=1
   12 PRINT 15,NBAD
      RETURN
   15 FORMAT(1H0,10X,30HRETURN FROM SBLMACH WITH NBAD=,I2,
     A39H BECAUSE ARGUMENTS ARE NOT WITHIN RANGE)
   25 IF(RZ.EQ.0..AND.TZ.GT.0.)GOTO 35
      IF(RZ.GT.0..AND.TZ.EQ.0.)GOTO 45
      NBAD=2
      GOTO 12
C  IN THIS CASE THE INITIAL NODE IS AT THE EXPLOSION
   35 TRAT=(Z1(V(1))*Z2V1)/(Z1(V(2))*Z2(V(2)))
      NUP=1
      IF(TRAT.GT.1.)NUP=-1
      ANUP=TZ/(Z1(V(1))*Z2V1**NUP)
      TBUP=TZ
      TFUP=AMUP*Z1(V(2))*Z2(V(2))**NUP
      RFUP=RSF(TFUP)
      AMDOWN=TZ/(Z1(V(1))*Z2V1**(-NUP))
      TBDOWN=TZ
      TFDOWN=AMDOWN*Z1(V(2))*Z2(V(2))**(-NUP)
      RFDOWN=RSF(TFDOWN)
C   THIS ESTABLISHED END POINTS OF BOTH CHARACTERISTICS.
C   NOW GO TO FIND OUT WHICH SEGMENT SHOULD BE COMPUTED
      GOTO 55
   45 TS=SQRT(ROC/CHEN)*(RZ/AK)**(0.5*FLOAT(2+NC))
C   IN THIS CASE A POINT ON THE SHOCK IS SPECIFIED
      TRAT=(Z1(V(1))*Z2V1)/(Z1(V(2))*Z2(V(2)))
      NUP=1
      IF(TRAT.GT.1.)NUP=-1
      AMUP=TS/(Z1(V(2))*Z2(V(2))**NUP)
      TBUP=AMUP*Z1(V(1))*Z2V1**NUP
      TFUP=TS
      RFUP=RZ
      AMDOWN=TS/(Z1(V(2))*Z2(V(2))**(-NUP))
      TBDOWN=AMDOWN*Z1(V(1))*Z2V1**(-NUP)
      TFDOWN=TS
      RFDOWN=RZ
C   AT 55 START WORK ON UPWARD CHARACTERISTIC
   55 IF(NR(1).LE.0)GOTO 205
C   BRANCH TO DOWNWARD CHARACTERISTIC
      IF(RFUP.LT.RMIN.OR.TFUP.LT.TMIN)GOTO 65
      IF(0..GT.RMAX.OR.TBUP.GT.TMAX)GOTO 65
      GOTO 75
   65 NR(1)=0
      GOTO 205
C   NOW FIND INTERSECTION WITH RMIN
   75 X1=0.
      F1=0.
      X2=1.0
      T2=TFUP
      F2=RFUP
      ICNT=0
      RM=AMAX1(RMIN,0.)
   85 X3=X2-(X2-X1)*(F2-RM)/(F2-F1)
      X3=AMAX1(0.,X3)
      IF(X3.LE.0.)T3=AMUP*Z1(XF(X3))*Z2V1**NUP
      IF(X3.GT.0.)T3=AMUP*Z1(XF(X3))*Z2(XF(X3))**NUP
      F3=RSF(T3)*Y(XF(X3))
      IF(ABS(F3-RM).LT.1.0E-7*RFDOWN) GOTO 95
      ICNT=ICNT+1
      IF(ICNT.GT.20)GOTO 95
      IF(F3-RM) 86,95,88
   86 X1=X3
      F1=F3
      T1=T3
      GOTO 85
   88 X2=X3
      F2=F3
      T2=T3
      GOTO 85
   95 X3=XF(X3)
```

```
      IF(T3-TMAX) 115,105,65
  105 NR(1)=1
      GOTO 135
  115 IF(T3.GE.TMIN) GOTO 135
C   BRANCH IF FIRST NODE WAS FOUND. ELSE GET INTERSECTION WITH TMIN
      X1=X3
      F1=T3
      X2=V(2)
      F2=TFUP
      T2=TFUP
      ICNT=0
  125 X3=X2-(X2-X1)*(F2-TMIN)/(F2-F1)
      T3=ANUP*Z1(X3)*Z2(X3)**NUP
      F3=T3
      IF(ABS(F3-TMIN).LT.1.0E-7*TBUP) GOTO 135
      ICNT=ICNT+1
      IF(ICNT.GT.20)GOTO 135
      IF(F3-TMIN) 126,135,128
  126 X1=X3
      T1=T3
      F1=F3
      GOTO 125
  128 X2=X3
      T2=T3
      F2=F3
      GOTO 125
  135 VIN=X3
      RS=RSF(T3)
      T(1,1)=T3
      R(1,1)=RS*Y(VIN)
      IF(R(1,1).GT.RMAX)GOTO 65
      USHCK=(2./FLOAT(2+NC))*RS/T(1,1)
      UP(1,1)=(R(1,1)/T(1,1))*VIN
      P(1,1)=(2./(GAM+1.))*ROC*USHCK**2*G(VIN)
      RO(1,1)=((GAM+1.)/(GAM-1.))*ROC*H(VIN)
      DP(1,1)=RO(1,1)*UP(1,1)**2*0.5
      IF(NR(1).EQ.1)GOTO 205
C   BRANCH TO COMPUTATION OF DOWNWARD CHARACTERISTIC
C   NOW FIND END POINT OF CURVE
      IF(R(1,1).EQ.RMAX)GOTO 177
      X1=((VIN-V(1))/(V(2)-V(1)))**C(2)
      F1=R(1,1)
      X2=1.
      T2=TFUP
      F2=RFUP
      ICNT=0
      RM=AMIN1(RMAX,RFUP)
  145 X3=X2-(X2-X1)*(F2-RM)/(F2-F1)
      T3=ANUP*Z1(XF(X3))*Z2(XF(X3))**NUP
      F3=RSF(T3)*Y(XF(X3))
      IF(ABS(F3-RM).LT.1.0E-7*RFDOWN) GOTO 155
      ICNT=ICNT+1
      IF(ICNT.GT.20)GOTO 155
      IF(F3-RM) 146,155,148
  146 X1=X3
      T1=T3
      F1=F3
      GOTO 145                                                          SBLMA172
  148 X2=X3                                                             SBLMA173
      T2=T3                                                             SBLMA174
      F2=F3                                                             SBLMA175
      GOTO 145                                                          SBLMA176
  155 X3=XF(X3)                                                         SBLMA177
      IF(T3.LE.TMAX) GOTO 175                                           SBLMA178
C  BRANCH IF END OF CURVE FOUND. ELSE FIND INTERSECTION WITH TMAX       SBLMA179
      X1=VIN                                                            SBLMA180
      F1=T(1,1)                                                         SBLMA181
      X2=X3                                                             SBLMA182
      F2=T3                                                             SBLMA183
      ICNT=0                                                            SBLMA184
  165 X3=X2-(X2-X1)*(F2-TMAX)/(F2-F1)                                   SBLMA185
      T3=AMUP*Z1(X3)*Z2(X3)**NUP                                        SBLMA186
      F3=T3                                                             SBLMA187
      IF(ABS(F3-TMAX).LT.1.0E-7 *TMAX)GOTO 175                          SBLMA188
      ICNT=ICNT+1                                                       SBLMA189
      IF(ICNT.GT.20)GOTO 175                                            SBLMA190
      IF(F3-TMAX) 166,175,168                                           SBLMA191
  166 X1=X3                                                             SBLMA192
      T1=T3                                                             SBLMA193
      F1=F3                                                             SBLMA194
      GOTO 165                                                          SBLMA195
  168 X2=X3                                                             SBLMA196
      T2=T3                                                             SBLMA197
      F2=F3                                                             SBLMA198
      GOTO 165                                                          SBLMA199
  175 VEN=X3                                                            SBLMA200
      IF(VEN.GT.VIN)GOTO 185                                            SBLMA201
  177 NR(1)=1                                                           SBLMA202
      GOTO 205                                                          SBLMA203
  185 KUP=NR(1)                                                         SBLMA204
      XIN=((VIN-V(1))/(V(2)-V(1)))**C(2)                                SBLMA205
      XEN=((VEN-V(1))/(V(2)-V(1)))**C(2)                                SBLMA206
      DO 195 KA=2,KUP                                                   SBLMA207
      X=XIN+(XEN-XIN)*FLOAT(KA-1)/FLOAT(KUP-1)                          SBLMA208
      X=XF(X)                                                           SBLMA209
      X=AMAX1(X,V(1))                                                   SBLMA210
      IF(X.LE.V(1)) T(1,KA)=AMUP*Z1(V(1))*Z2V1**NUP                     SBLMA211
      IF(X.GT.V(1)) T(1,KA)=AMUP*Z1(X)*Z2(X)**NUP                       SBLMA212
      RS=RSF(T(1,KA))                                                   SBLMA213
      R(1,KA)=RS*Y(X)                                                   SBLMA214
      USHCK=(2./FLOAT(2+NC))*RS/T(1,KA)                                 SBLMA215
      P(1,KA)=(2./(GAM+1.))*ROC*USHCK**2*G(X)                           SBLMA216
      UP(1,KA)=X*R(1,KA)/T(1,KA)                                        SBLMA217
      RO(1,KA)=((GAM+1.)/(GAM-1.))*ROC*H(X)                             SBLMA218
      DP(1,KA)=RO(1,KA)*UP(1,KA)**2*0.5                                 SBLMA219
  195 CONTINUE                                                          SBLMA220
C  AT 205 START COMPUTATION OF DOWNWARD CURVE                           SBLMA221
  205 IF(NR(2).LE.0)RETURN                                              SBLMA222
      IF(RFDOWN.LT.RMIN.OR.TFDOWN.GT.TMAX)GOTO 215                      SBLMA223
      IF(0..GT.RMAX.OR.TBDOWN.LT.TMIN)GOTO 215                          SBLMA224
      GOTO 225                                                          SBLMA225
  215 NR(2)=0                                                           SBLMA226
      RETURN                                                            SBLMA227
C  AT 225 FIND DOWNWARD INTERSECTION WITH RMIN                          SBLMA228
```

```
  225 X1=0.0
      F1=0
      X2=1.0
      T2=TFDOWN
      F2=RFDOWN
      ICNT=0
      RM=AMAX1(RMIN,0.)
  235 X3=X2-(X2-X1)*(F2-RM)/(F2-F1)
      X3=AMAX1(0.,X3)
      IF(X3.LE.0.0)T3=AMDOWN*Z1(XF(X3))/Z2V1**NUP
      IF(X3.GT.0.0)T3=AMDOWN*Z1(XF(X3))/Z2(XF(X3))**NUP
      F3=RSF(T3)*Y(XF(X3))
      IF(ABS(F3-RM).LT.1.0E-7 *RFDOWN)GOTO 245
      ICNT=ICNT+1
      IF(ICNT.GT.20)GOTO 245
      IF(F3-RM) 236,245,238
  236 X1=X3
      T1=T3
      F1=F3
      GOTO 235
  238 X2=X3
      T2=T3
      F2=F3
      GOTO 235
  245 X3=XF(X3)
      IF(T3-TMIN)215,255,265
  255 NR(2)=1
      GOTO 285
  265 IF(T3.LE.TMAX)GOTO 285
C  BRANCH IF FIRST NODE FOUND
      X1=X3
      F1=T3
      X2=V(2)
      T2=TFDOWN
      F2=T2
      ICNT=0
  275 X3=X2-(X2-X1)*(F2-TMAX)/(F2-F1)
      T3=AMDOWN*Z1(X3)/Z2(X3)**NUP
      F3=T3
      IF(ABS(F3-TMAX).LT.1.0E-7 *TBDOWN)GOTO 285
      ICNT=ICNT+1
      IF(ICNT.GT.20)GOTO 285
      IF(F3-TMAX) 276,285,278
  276 X2=X3
      T2=T3
      F2=F3
      GOTO 275
  278 X1=X3
      T1=T3
      F1=F3
      GOTO 275
  285 VIN=X3
      RS=RSF(T3)
      T(2,1)=T3
      R(2,1)=RS*Y(VIN)
      IF(R(2,1).GT.RMAX)GOTO 215
      USHCK=(2./FLOAT(2+NC))*RS/T(2,1)
      P(2,1)=(2.0/(GAM+1.))*ROC*USHCK**2*G(VIN)                         SBLMA286
      UP(2,1)=VIN*R(2,1)/T(2,1)                                         SBLMA287
      RO(2,1)=((GAM+1.)/(GAM-1.))*ROC*H(VIN)                            SBLMA288
      DP(2,1)=RO(2,1)*UP(2,1)**2*0.5                                    SBLMA289
      IF(R(2,1).EQ.RMAX)NR(2)=1                                         SBLMA290
      IF(NR(2).EQ.1)RETURN                                              SBLMA291
C   NOW FIND END POINT OF DOWNWARD CURVE                                SBLMA292
C   FIRST FIND INTERSECTION WITH RMAX                                   SBLMA293
      X1=((VIN-V(1))/(V(2)-V(1)))**C(2)                                 SBLMA294
      F1=R(2,1)                                                         SBLMA295
      X2=1.0                                                            SBLMA296
      T2=TFDOWN                                                         SBLMA297
      F2=RFDOWN                                                         SBLMA298
      ICNT=0                                                            SBLMA299
      RM=AMIN1(RFDOWN,RMAX)                                             SBLMA300
  295 X3=X2-(X2-X1)*(F2-RM)/(F2-F1)                                     SBLMA301
      T3=ANDOWN*Z1(XF(X3))/Z2(XF(X3))**NUP                              SBLMA302
      F3=RSF(T3)*Y(XF(X3))                                              SBLMA303
      IF(ABS(F3-RM).LT.1.0E-7 *RFDOWN)GOTO 305                          SBLMA304
      ICNT=ICNT+1                                                       SBLMA305
      IF(ICNT.GT.20)GOTO 305                                            SBLMA306
      IF(F3-RM) 296,305,298                                             SBLMA307
  296 X1=X3                                                             SBLMA308
      T1=T3                                                             SBLMA309
      F1=F3                                                             SBLMA310
      GOTO 295                                                          SBLMA311
  298 X2=X3                                                             SBLMA312
      T2=T3                                                             SBLMA313
      F2=F3                                                             SBLMA314
      GOTO 295                                                          SBLMA315
  305 X3=XF(X3)                                                         SBLMA316
      IF(T3.GE.TMIN)GOTO 325                                            SBLMA317
C   BRANCH IF END POINT FOUND. ELSE FIND INTERSECTION WITH TMIN         SBLMA318
      X1=VIN                                                            SBLMA319
      F1=T(2,1)                                                         SBLMA320
      F2=T3                                                             SBLMA321
      X2=X3                                                             SBLMA322
      ICNT=0                                                            SBLMA323
  315 X3=X2-(X2-X1)*(F2-TMIN)/(F2-F1)                                   SBLMA324
      T3=ANDOWN*Z1(X3)/Z2(X3)**NUP                                      SBLMA325
      F3=T3                                                             SBLMA326
      IF(ABS(F3-TMIN).LT.1.0L-7 *TBDOWN)GOTO 325                        SBLMA327
      ICNT=ICNT+1                                                       SBLMA328
      IF(ICNT.GT.20)GOTO 325                                            SBLMA329
      IF(F3-TMIN) 316,325,318                                           SBLMA330
  316 X2=X3                                                             SBLMA331
      T2=T3                                                             SBLMA332
      F2=F3                                                             SBLMA333
      GOTO 315                                                          SBLMA334
  318 X1=X3                                                             SBLMA335
      T1=T3                                                             SBLMA336
      F1=F3                                                             SBLMA337
      GOTO 315                                                          SBLMA338
  325 VEN=X3                                                            SBLMA339
      IF(VEN.GT.VIN)GOTO 335                                            SBLMA340
      NR(2)=1                                                           SBLMA341
      RETURN                                                            SBLMA342
```

```
C   LOOP TO COMPUTE DOWNWARD CHARACTERISTIC                              SBLMA343
  335 KUP=NR(2)                                                          SBLMA344
      XIN=((VIN-V(1))/(V(2)-V(1)))**C(2)                                 SBLMA345
      XEN=((VEN-V(1))/(V(2)-V(1)))**C(2)                                 SBLMA346
      DO 345 KA=2,KUP                                                    SBLMA347
      X=XIN+(XEN-XIN)*FLOAT(KA-1)/FLOAT(KUP-1)                           SBLMA348
      X=XF(X)                                                            SBLMA349
      X=AMAX1(X,V(1))                                                    SBLMA350
      IF(X.LE.V(1)) T(2,KA)=AMDOWN*Z1(V(1))/Z2V1**NUP                    SBLMA351
      IF(X.GT.V(1)) T(2,KA)=AMDOWN*Z1(X)/Z2(X)**NUP                      SBLMA352
      RS=RSF(T(2,KA))                                                    SBLMA353
      R(2,KA)=RS*Y(X)                                                    SBLMA354
      USHCK=(2./FLOAT(2+NC))*RS/T(2,KA)                                  SBLMA355
      P(2,KA)=(2./(GAM+1.))*ROC*USHCK**2*G(X)                            SBLMA356
      UP(2,KA)=X*R(2,KA)/T(2,KA)                                         SBLMA357
      RO(2,KA)=((GAM+1.)/(GAM-1.))*ROC*H(X)                              SBLMA358
      DP(2,KA)=RO(2,KA)*UP(2,KA)**2*0.5                                  SBLMA359
  345 CONTINUE                                                           SBLMA360
      RETURN                                                             SBLMA361
      END                                                                SBLMA362
```

DISTRIBUTION LIST

No. of Copies Organization

12 Commander
Defense Technical Info Center
ATTN: DDC-DDA
Cameron Station
Alexandria, VA 22314

2 Director
Defense Civil Preparedness
Agency
ATTN: Mr. George Sission/RF-SR
Technical Library
Washington, DC 20301

1 Director
Defense Advanced Research
Projects Agency
ATTN: Technical Library
1400 Wilson Boulevard
Arlington, VA 22209

10 Director
Defense Nuclear Agency
ATTN: DDST/Dr. Conrad
DDST/Dr. Oswald
DDST/Dr. Knowles
SPSS/Dr. Goering
SPSS/Dr. Ullrich
STSI/Archives
STSP
STTL/Tech Lib (2 cys)
STVL
Washington, DC 20305

5 Commander
US Army Engineer Waterways
Experiment Station
ATTN: Technical Library
William Flathau
Leo Ingram
Guy Jackson
John N. Strange
P.O. Box 631
Vicksburg, MS 39180

No. of Copies Organization

1 Commander
US Army Materiel Development
& Readiness Command
ATTN: DRCDMD-ST
5001 Eisenhower Avenue
Alexandria, VA 22333

1 Commander
US Army Materiel Development
& Readiness Command
ATTN: Technical Library
5001 Eisenhower Avenue
Alexandria, VA 22333

2 Commander
US Army Armament Research
& Development Command
ATTN: DRDAR-TSS (2 cys)
Dover, NJ 07801

2 Commander
US Army Armament Research
& Development Command
ATTN: DRDAR-LCN, P. Angelotti (2 cys)
Dover, NJ 07801

1 Commander
US Army Armament Materiel
Readiness Command
ATTN: DRSAR-LEP-L, Tech Lib
Rock Island, IL 61299

1 Director
US Army ARRADCOM
Benet Weapons Laboratory
ATTN: DRDAR-LCB-TL
Watervliet, NY 12189

1 Commander
US Army Aviation Research
& Development Command
ATTN: DRSAV-E
P.O. Box 209
St. Louis, MO 63166

DISTRIBUTION LIST

No. of Copies Organization

1 Director
US Army Air Mobility Research
& Development Laboratory
Ames Research Center
Moffett Field, CA 94035

1 Commander
US Army Communications Research
& Development Command
ATTN: DRDCO-PPA-SA
Fort Monmouth, NJ 07703

1 Commander
US Army Electronics Research
& Development Command
Technical Support Activity
ATTN: DELSD-L
Fort Monmouth, NJ 07703

10 Commander
US Army Harry Diamond Labs
ATTN: DRXDO-TI
DRXDO-TI/012
DRXDO-NP
DRXDO-RBH, Mr. Caldwell
DELHD-RBA, J. Rosado
Mr. L. Belliveau
Mr. James Gaul
Mr. J. Gwaltney
Mr. Bill Vault
Mr. F.N. Wimenitz
2800 Powder Mill Road
Adelphi, MD 20783

1 Commander
US Army Missile Command
ATTN: DRSMI-R
Redstone Arsenal, AL 35809

1 Commander
US Army Missile Command
ATTN: DRSMI-YDL
Redstone Arsenal, AL 35809

No. of Copies Organization

1 Commander
US Army Tank Automotive Research
& Development Command
ATTN: DRDTA-UL
Warren, MI 48090

4 Commander
US Army Nuclear Agency
ATTN: ACTA-NAW
Technical Library
Dr. Davidson
MAJ J. Sims
7500 Backlick Road, Bldg 2073
Springfield, VA 22150

1 Director
US Army TRADOC Systems
Analysis Activity
ATTN: ATAA-SL, Tech Lib
White Sands Missile Range,
NM 88002

1 Director
Strategic Systems Project Office
ATTN: NSP-43, Tech Lib
Department of the Navy
Washington, DC 20360

1 Commander
Naval Surface Weapons Center
ATTN: DX-21, Lib Br
Dahlgren, VA 22448

2 Commander
Naval Surface Weapons Center
ATTN: Code WX21, Tech Lib
Code 240, C.J. Aronson
Silver Spring, MD 20910

1 AFWL/DE-I
Kirtland AFB, NM 87117

1 AFWL/DEX
Kirtland AFB, NM 87117

DISTRIBUTION LIST

No. of Copies — Organization

1 AFWL/SUL, Jimmie L. Bratton
Kirtland AFB, NM 87117

1 AFWL, R. Henny
Kirtland AFB, NM 87117

1 AFWL/SUL, M.A. Plamondon
Kirtland AFB, NM 87117

1 Director
Lawrence Livermore Laboratory
ATTN: Tech Info Dept L-3
P.O. Box 808
Livermore, CA 94550

4 Director
Los Alamos Scientific Laboratory
ATTN: Doc Contr for Rpts Lib
Al Davis
R.A. Gentry
G.R. Spillman
P.O. Box 1663
Los Alamos, NM 87544

3 Sandia Laboratories
ATTN: Doc Control for 3141
Sandia Rpt Collection
M.L. Merritt
L.J. Vortman
P.O. Box 5800
Albuquerque, NM 87115

1 Agbabian Associates
ATTN: M. Agbabian
250 North Nash Street
El Segundo, CA 90245

1 Analytical Services, Inc.
ATTN: George Hessellracher
712 F North Valley Street
Anaheim, CA 92801

1 The BMD Corporation
ATTN: Richard Hensely
P.O. Box 9274
Albuquerque International
Albuquerque, NM 87119

No. of Copies — Organization

1 General Electric Company-TEMPO
ATTN: DASIAC
P.O. Drawer QQ
Santa Barbara, CA 93102

1 General Electric Company-TEMPO
ATTN: E. Bryant
220 S. Main Street, Rm 206
Bel Air, MD 21014

2 Kaman Avidyne
ATTN: Dr. N.P. Hobbs
Mr. S. Criscione
83 Second Avenue
Northwest Industrial Park
Burlington, MA 01830

2 Kaman Sciences Corporation
ATTN: Library
F.H. Shelton
1500 Garden of the Gods Road
Colorado Springs, CO 80907

1 Kaman Sciences Corporation
ATTN: Don Sachs
Suite 855
101 Continental Boulevard
El Segundo, CA 90245

3 Pacific Sierra Research Corp.
ATTN: F.M. Sauer
J.M. Thomsen
Technical Library
2700 Merced Street
San Leandro, CA 94577

1 R&D Associates
ATTN: Henry Cooper
Suite 500
1401 Wilson Boulevard
Arlington, VA 22209

DISTRIBUTION LIST

No. of Copies   Organization

6 R&D Associates
ATTN: Jerry Carpenter
Allen L. Kuhl
Dr. Albert L. Latter
J.G. Lewis
William B. Wright
Technical Library
P.O. Box 9695
Marina del Rey, CA 90291

1 The Rand Corporation
ATTN: C.C. Mow
1700 Main Street
Santa Monica, CA 90406

2 Science Applications, Inc.
ATTN: John Cockayne
Wm. Layson
1710 Goodridge Dr.
P.O.Box 1303
McLean, VA 22101

1 Science Applications, Inc.
2450 Washington Avenue
San Leandro, CA 94577

2 Science Applications, Inc.
ATTN: Technical Library
Michael McKay
P.O. Box 2351
La Jolla, CA 92037

4 Systems, Science & Software
ATTN: Ted Cherry
Donald R. Grine
Thomas D. Riney
Technical Library
P.O. Box 1620
La Jolla, CA 92037

1 Systems, Science & Software
ATTN: Charles Needham
P.O. Box 8243
Albuquerque, NM 87198

No. of Copies   Organization

3 Terra Tek, Inc.
ATTN: Sidney Green
A.H. Jones
Technical Library
420 Wakara Way
Salt Lake City, UT 84108

2 TRW Systems Group
ATTN: Benjamin Sussholts
Tech Info Ctr/S-1930
One Space Park
Redonodo Beach, CA 92078

1 Weidlinger Associate
Consulting Engineers
ATTN: M.L. Baron
110 East 59th Street
New York, NY 10022

2 Drexel Institute of Technology
ATTN: Prof. P.C. Chou
Mechanics and Structures Group
32 and Chestnuts Streets
Philadelphia, PA 19137

1 Massachusetts Institute of Technology
Aeroelastic and Structures
Research Laboratory
ATTN: Dr. E.A. Witmer
Cambridge, MA 02139

2 Southwest Research Institute
ATTN: Dr. W.E. Baker
A.B. Wenzel
8500 Culebra Road
San Antonio, TX 78206

2 SRI International
ATTN: Dr. G.R. Abrahamson
Carl Peterson
333 Ravenswood Avenue
Menlo Park, CA 94025

DISTRIBUTION LIST

No. of Copies Organization

2 University of Denver
Denver Research Institute
ATTN: Mr. J. Wisotski
Technical Library
P.O. Box 10127
Denver, CO 80210

1 University of Illinois
Consulting Engineering Services
ATTN: Nathan M. Newmark
1211 Civil Engineering Building
Urbana, IL 61801

No. of Copies Organization

Aberdeen Proving Ground

Dir, USAMSAA
ATTN: DRXSY-D
DRXSY-MP, H. Cohen
Cdr, USATECOM
ATTN: DRSTE-TO-F
Dir, CSL
Bldg E3516, EA
ATTN: DRDAR-CLB-PA

USER EVALUATION OF REPORT

Please take a few minutes to answer the questions below; tear out this sheet and return it to Director, US Army Ballistic Research Laboratory, ARRADCOM, ATTN: DRDAR-TSB, Aberdeen Proving Ground, Maryland 21005. Your comments will provide us with information for improving future reports.

1. BRL Report Number ______________________________

2. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which report will be used.)

____

____

____

3. How, specifically, is the report being used? (Information source, design data or procedure, management procedure, source of ideas, etc.) ______________________________

____

____

4. Has the information in this report led to any quantitative savings as far as man-hours/contract dollars saved, operating costs avoided, efficiencies achieved, etc.? If so, please elaborate.

____

____

5. General Comments (Indicate what you think should be changed to make this report and future reports of this type more responsive to your needs, more usable, improve readability, etc.) ______________________________

____

____

____

6. If you would like to be contacted by the personnel who prepared this report to raise specific questions or discuss the topic, please fill in the following information.

Name: ______________________________

Telephone Number: ______________________________

Organization Address: ______________________________

____

____